The MICROSOFT® BITMAP® image format is a public-domain format supporting images in the Truecolor, color palette, grey-scale, or black and white representations. Truecolor images have 24-bits per pixel element, with each byte of the pixel element representing the intensity of the red, green, and blue (RGB) color component. Color palette images contain a table of the permitted RGB values. The pixel value in a color palette image is an index to this table. Grey-scale images give the numerical intensity of the pixel values. Black and white representation assigns either zero or one as one of the two possible pixel values. The invention will be made understandable in the context of the BITMAP® image types by reference to the following description.

At the point when most sensory obtained information is represented in digital form, whether it be from video, photographs, laboratory measurements, or facsimile transmissions, the digital data contain intrinsic noise and redundant information which can be manipulated to carry extra information. Through use of this invention, the extra information also can be extracted easily by an authorized and enabled receiver of the data.

Redundancy in digital image data occurs when a particular numerical value of the digital intensity exists in many different parts of the image. Redundancy is found commonly in images because a given grey-level exists in the rendition of trees, sky, clouds, people, and other objects. The presence of noise in digital images permits the picture elements, pixels, to vary slightly in numerical value. For

8-bit digital data, the pixel numerical values range from 0-255. As the pixels having the same or similar numerical values represent a variety of image content, many values in different locations of an image can be interchanged freely. The image appearance and the statistical frequency of occurrence of a particular pixel value are affected little by the interchanging of the spatial position of pixels close in numerical value.

Initially, from the original digital data (hereinafter often referred to as the "host" data), the present invention first converts the host data to digital form, if necessary, and then creates an image histogram to show the probability density of numerical pixel values occurring in the image. The number of times a particular pixel value occurs in the image is plotted versus the value. For 8-bit digital data, the pixel values range from 0-255. Of course, the level of noise in an image will depend on the source of the data, with different noise levels expected between photos, original artwork, digital audio, video, and facsimile transmissions.

The actual embedding of the auxiliary data into the host data is a three-part process, the basic steps of which are illustrated in Figure 1. First, an estimate of the noise component of the host data is determined and used in combination with an analysis of a histogram of the host data numerical values to identify pairs of values in the host data that occur with approximately the same statistical frequency, and that differ in value by less than the value of the noise component. Second, the position of occurrence

of the pair values found is adjusted to embed the bitstream of the auxiliary information set. Third, the identified pairs of values in the host data are used to create a key for the extraction of the embedded data.

Extracting embedded data inverts this process. The key placed in the image in the embedding phase specifies the pair-values which contain the embedded auxiliary information. With the pair-values known, extraction consists of recreating the auxiliary data according to the positions of pixels having the pair-values given in the key. The key data are used first to extract header information. The header information specifies the length and the file name of the auxiliary data, and serves to validate the key. If the image containing embedded information has been modified, the header information will not extract data correctly. However, successful extraction recreates the auxiliary data exactly in an output file.

The principle of data embedding according to the present invention involves the rearrangement of certain host data values in order to encode the values of the extra data which is to be added. For the purposes of this description of the invention, consider a host data set represented by 8 bits of binary information, with values ranging between 0 and 255 bits for each host data sample. Further, assume that the noise value, N, for a signal, S, is given by $N=\pm S/10$, or approximately 10% of the signal value. For many data, the noise component can be approximated by a constant value or percentage, such as the 10% value used for this

description. Two values in the host data, d_i and d_j, are within the noise value if:

$$|d_i - d_j| = \varepsilon \leq N \qquad 10$$

The frequency of occurrence or histogram value of a certain value, d_i, is $f(d_i)$. Data values meeting the criteria of Equation 10, and occurring in the host data with frequency of occurrence $f(d_i)-f(d_j)< \delta$, where δ is the tolerance imposed for statistical equality, are candidates for embedding use. The values, d_i and d_j, constitute a pair of data values, p_k. There are $k=0,1,2,...N_p$ such pairs in the host data set, giving a total number of embedding bits, M_k, for each pair:

$$M_k = \sum_i f(d_i) + \sum_j f(d_j) \qquad 20$$

where the summations for i and j run to the limits of the frequency of occurrence in the data set, $f(d_i)$ and $f(d_j)$, for the respective data values.

It is now helpful to refer to Figure 2, wherein a partial listing of computer code in the C-Language is printed. The determination of the host data pixel pair values, d_i and d_j, in Equation 10, is accomplished through the code listed in Figure 2. In Figure 2, these 8 bit values are interpreted as indices in a color palette table. The comparison indicated in Equation 10 is therefore required to be a comparison between the corresponding colors

in the palette. Entries in the color palette are Red, Green, and Blue (RGB) color-component values, each within the range of 0-255.

If additional information is desired on the format used for BITMAP® images, reference should be made to two sources. One is the book, ***Programming for Graphics Files***, by J. Levine, 1994 (J. Wiley & Sons, New York). The other is a technical article, "The BMP Format," by M. Luse, ***Dr. Dobb's Journal***, Vol. 19, Page 18, 1994.

The code fragment in Figure 2 begins at line 1 with a loop running over the number of colors in the palette. The loop index, i, is used to test each palette color against all other entries, in sequence, to identify pairs of color entries meeting the criteria established by Equation 10. Each color identified in the i-loop then is tested against all other colors in the palette by a second loop using another index, j, starting at line 16. Line 7 provides a modification for images which have a palette for greyscale instead of colors. For greyscale images, the RGB components are identical for each palette entry, although some grey scale formats include a 16-color table as well.

The comparison indicated in Equation 10 is made by converting the Red, Green, and Blue (RGB) color component values to corresponding Hue, Saturation, and Intensity (HSI) color components. Line 12 uses a separate routine, rgbhsi(), to effect this conversion. Line 20 converts RGB color component values in the j-loop to HSI data structure components, and line 21 calculates the color difference in the HSI system. Line 24 then implements the test required

by Equation 10. If the color difference is less than a fixed noise value (COLOR_NOISE=10 in the listing of Figure 2), the intensity difference is tested to determine if the two palette entries are acceptable as differing by less than the noise value specified. Two additional constraints are imposed before accepting the entries as candidate pair values. First, the difference in color is required to be the smallest color difference between the test (i-loop) value, and all the other (j-loop) values. Second, the number of pairs selected (k) must be less than half the number of columns in a row of pixels in the image, in order for the pair-value key to be stored in a single row of pixels. This is an algorithmic constraint, and is not required by the invention.

A data-structure array, pair[], is used to hold the values of candidate pairs (i,j) and their total frequency of occurrence, M_k. If the image is a greyscale palette, the test at line 35 is used to force comparison of only the intensity of the two palette entries. Greyscale images do not require the RGB to HSI conversion made for color palettes.

The embedding process of the present invention ignores differences in the saturation component of color palette entries because saturation is ordinarily not noticeable in a color image. Only the Hue and Intensity components are constrained to fall within fixed noise limits to determine the palette pair values.

Pixel pair values found by the code listed in Figure 2 include generally redundant values. The same pixel value,

i, is found in several different pair combinations. Because multiple pairs cannot contain the same palette entry, due to each pair combination of pixel values having to be unique, it is necessary to eliminate some pairs. The number of pairs located by applying the criterion of Equation 10 is stored in the variable, no_pairs, in line 51.

Referring now to Figure 3, the code fragment listed therein illustrates the manner in which duplicate pairs are eliminated by a separate routine. First, the histogram of the image is used to calculate the total number of occurrences in each pair, as required by Equation 20, above. Line 1 shows the i-loop used to calculate the value, M_k, for each pair. Next, the pairs are sorted according to decreasing order of the pair[].count data-structure member in line 5. The elimination of duplicates in the following line retains the pairs, p_k, having the largest total number of frequency values, M_k. Line 10 and the lines following calculate the total number of bytes that can be embedded into the host data using the unique pixel pairs found by this code fragment.

Sorting the pair values in decreasing order of value, M_k, minimizes the number of pairs required to embed a particular auxiliary data stream. However, the security of the embedded data is increased significantly if the pair values are arranged in random order. Randomizing the pair-value order is part of this invention. This is accomplished by rearranging the pair-values to random order by calculating a data structure having entries for an integer index pts[k].i, k=0,1,2,..., no_pairs; and pts[k].gamma=δ_0,

$\delta_1, \ldots \delta_r, \ldots \delta_{no\_pairs}$, where the δ_r values are random. Sorting the data structure, pts[], to put the random values in ascending order randomizes the index values. The random index values are used with the pair-values calculated as indicated above, to re-order the table to give random pair-value ordering.

The algorithm described for palette-format images permits manipulating pixel values without regard to the individual frequency of occurrence. Reference should now be made to Figure 4 where another code fragment is listed in which, for Truecolor images, a constraint is introduced on the frequency of occurrence that minimizes the effect of embedding on the host data histogram.

Truecolor images consist of three individual 8-bit greyscale images, one each for the red, green, and blue image components. Truecolor images have no color palette. The possible combinations of the three 8-bit components give approximately 16 million colors. The present invention embeds data into Truecolor images by treating each RGB color component image separately. The effect of embedding on the composite image color is therefore within the noise value of the individual color intensity components.

In Figure 4, the ip-loop starting in line 2 refers to the color plane (ip=0,1,2 for R,G,B). The frequency of occurrence of each numerical value (0 through 255) is given in the array, hist_values[], with the color plane histograms offset by the quantity, ip*256, in line 7. The variable, fvalue[], holds the floating point histogram values for color-component, ip. Line 11 begins a loop to constrain the

pairs selected for nearly equal frequency of occurrence. Pixel intensities within the noise limit, RANGE, are selected for comparison of statistical frequency. The tolerance, δ, for statistical agreement is fixed at 5% in line 17. This tolerance could be adjusted for particular applications.

After all possible values are tested for the constraints of noise and statistical frequency, the pairs found are sorted in line 27, the duplicates are removed, the starting index is incremented in line 31, and the search continued. A maximum number of pairs again is set by the algorithmic constraint that the i and j pair values must be less than one-half the number of pixels in an image row. As with palette-format images, the security of the invention includes randomizing the pair-value entries.

Applying the statistical constraint minimizes the host image effects of embedding the auxiliary data. If the tolerance, δ, is set at 0, each pair chosen will contain data values less than the noise value in intensity separation, and occurring with exactly the same statistical frequency. Setting the tolerance at δ=5%, as in the code fragment of Figure 4, permits the acceptance of pixel pairs that are close in frequency, while still preserving most of the statistical properties of the host data. Few, if any, pairs might be found by requiring exactly the same frequency of occurrence.

The actual embedding of auxiliary data into a set of host data consists of rearranging the order of occurrence of redundant numerical values. The pairs of host data values

found by analysis are the pixel values used to encode the bit-stream of the auxiliary data into the host data. It is important to realize that the numerical values used for embedding are the values already occurring in the host data. The embedding process of the current invention does not alter the number or quantity of the numerical values in the host data.

In the embedding process of the present invention, the host data are processed sequentially. A first pass through the host data examines each value and tests for a match with the pixel-pair values. Matching values in the host data are initialized to the data-structure value, pair[k].i, for $k=0,1,2...N_p$. This step initializes the host BITMAP® image (Figure 1) to the pair values corresponding to zeroes in the auxiliary data. A second pass through the auxiliary data examines the sequential bits of the data to be embedded, and sets the pair-value of the host data element to the value i or j, according to the auxiliary bit value to be embedded. If the bit-stream being embedded is random, the host data pair-values, i and j, occur with equal frequency in the host image after the embedding process is completed.

Figure 5 illustrates the code fragment that performs the actual embedding, including the considerable information which is necessary to manipulate the data in the header information, auxiliary bit-stream, and the host data files. Lines 1-12 allocate memory and initialize variables. The header and bit-stream data to be embedded are denoted the "data-image," and are stored in the array, data_row[]. The host data are denoted the "image-data."

The index, li, is used in a loop beginning at line 12 to count the byte position in the data-image. The loop begins with li=-512 because header information is embedded before the data-image bytes. Line 14 contains the test for loading data_row[] with the header information. Line 20 contains the test for loading data_row[] with bytes from the data-image file, tape5.

Line 30 starts a loop for the bits within a data-image byte. The variable, bitindex=(0,1,2...7), counts the bit position within the data-image byte, data_row[d_inrow], indexed by the variable, d_inrow. The variable, lj, indexes the byte (pixel) in the host image. The variable, inrow, indexes the image-data buffer, image_row[inrow]. Line 32 tests for output of embedded data (a completed row of pixels) to the image-data file, and line 40 tests for completion of a pass through the image-data. One pass through the image-data is made for each of the pixel pairs, pair[k], k=0,1,2...N_p.

In line 47, the pair index is incremented. A temporary pair data-structure variable named "pvalue" is used to hold the working pair values of the host data pixels being used for embedding. Line 60 provides for refreshing the image-data buffer, image_row.

The embedding test is made at line 72. If the image_row[inrow] content equals the pair value representing a data-image bit of zero, no change is made, and the image-data value remains pvalue.i. However, if the bit-stream value is one, the image-data value is changed to equal pvalue.j. Line 84 treats the case for image-data values not

equal to the embedding pair value, pvalue.i. In this case, the bitindex variable is decremented, because the data-image bit is not yet embedded, and the image-data indices are incremented to examine the next host data value.

The extraction of embedded data is accomplished by reversing the process used to embed the auxiliary data-image bit-stream. A histogram analysis of the embedded image-data set will reveal the candidate pairs for extraction for only the case where the individual statistical frequencies are unchanged by the embedding process. In the listings of Figures 2-5, the statistical frequencies are changed slightly by the embedding process. The pair table used for embedding can be recreated by analysis of the original image-data, but it cannot generally be recovered exactly from the embedded image-data.

Additionally, as described above, the invention includes randomizing the order of the pair-values, thereby increasing greatly the amount of analysis needed to extract the embedded data without prior knowledge of the pair-value order.

As previously described, the ordered pairs selected for embedding constitute the "key" for extraction of the data-image from the image-data. The listings illustrated in Figures 2-5 demonstrate how embedding analysis reduces the statistical properties of the noise component in host data to a table of pairs of numerical values. The key-pairs are required for extraction of the embedded data, but they cannot be generated by analyzing the host data after the embedding process is completed. However, the key can be

recreated from the original, unmodified host data. Thus, data embedding is similar to one-time-pad encryption, providing extremely high security to the embedded bit-stream.

With the pair table known, extraction consists of sequentially testing the pixel values to recreate an output bit-stream for the header information and the data-image. In the present invention, the pair table is inserted into the host image-data, where it is available for the extraction process. Optionally, the present invention permits removing the pair table, and storing it in a separate file. Typically, the pair table ranges from a few to perhaps hundreds of bytes in size. The maximum table size permitted is one-half the row length in pixels. With the pair table missing, the embedded data are secure, as long as the original host image-data are unavailable. Thus, the embedding method gives security potential approaching a one-time-pad encryption method.

Another way of protecting the pair table is to remove the key and encrypt it using public-key or another encryption process. The present invention permits an encrypted key to be placed into the host image-data, preventing extraction by unauthorized persons.

Embedding auxiliary data into a host slightly changes the statistical frequency of occurrence of the pixel values used for encoding the bit-stream. Compressed or encrypted embedding data are excellent pseudo-random auxiliary bit-streams. Consequently, embedding auxiliary data having pseudo-random properties minimizes changes in the average

frequency of occurrence of the values in the embedding pairs. Embedding character data without compression or encryption reduces significantly the security offered by the present invention.

The existence of embedded data is not detected easily by analyzing the embedded image-data. When viewed as a cryptographic method, data embedding convolves the data-image with the image-data. The original data-image bit-stream embedded represents a plaintext. The combination of the host and embedded data implants ciphertext in the noise component of the host. The existence of ciphertext is not evident however, because the content and meaning of the host carrier information is preserved by the present invention. Data embedding according to the present invention is distinct from encryption because no obvious ciphertext is produced.

Those who are unfamiliar with the terms "plaintext," and "ciphertext" can refer, for example, to B. Schneier, ***Applied Cryptography Protocols, Algorithms, and Source Code in C***, J. Wiley & Sons, New York, New York, 1994. This reference is incorporated herein by reference.

As mentioned previously, the present invention is useful in the embedding of auxiliary information into facsimile (FAX) data. In the previous discussion concerning embedding auxiliary information into image host data, the noise component originates from uncertainty in the numerical values of the pixel data, or in the values of the colors in a color pallet.

Facsimile transmissions are actually images consisting of black and white BITMAP® data, that is, the data from image pixels are binary (0,1) values representing black or white, respectively, and the effect of noise is to either add or remove pixels from the data. The present invention, therefore, processes a facsimile black-and-white BITMAP® image as a 2-color BITMAP®.

The standard office FAX machine combines the scanner and the digital hardware and software required to transmit the image through a telephone connection. The images are transmitted using a special modem protocol, the characteristics of which are available through numerous sources. One such source, the ***User's Manual for the EXP Modem*** (UM, 1993), describes a FAX/data modem designed for use in laptop computers. FAX transmissions made between computers are digital communications, and the data are therefore suited to data embedding.

As has been previously discussed with relation to embedding into images, the FAX embedding process is conducted in two stages: analysis and embedding. In the case of a FAX 2-color BITMAP®, image noise can either add or subtract black pixels from the image. Because of this, the length of runs of consecutive like pixels will vary.

The scanning process represents a black line in the source copy by a run of consecutive black pixels in the two color BITMAP® image. The number of pixels in the run is uncertain by at least ±1, because of the scanner resolution and the uncertain conversion of original material to black-and-white BITMAP® format.

Applying data embedding to the two color BITMAP® data example given here therefore consists of analyzing the BITMAP® to determine the statistical frequency of occurrence, or histogram, of runs of consecutive pixels. The embedding process of the present invention varies the length of runs by (0,+1) pixel according to the content of the bit-stream in the auxiliary data- image. Host data suitable for embedding are any two color BITMAP® image which is scaled in size for FAX transmission. A hardcopy of a FAX transmission can be scanned to generate the two color BITMAP®, or the image can be created by using FAX-printer driver software in a computer.

The FAX embedding process begins by analyzing the lengths of runs in each row of pixels. The implementation of this step is illustrated by the code fragment in Figure 6. The arguments to the routine, rowstats() are a pointer to the pixel data in the row, which consists of one byte per pixel, either a zero or a one in value; a pointer to an array of statistical frequencies; the number of columns (pixels) in the data row; and a flag for internal program options. The options flag is the size of blocks, or packets, of the auxiliary bitstream to be embedded. The options flag is tested in line 9, and the routine, packet_col() is used for a positive option flag. The packet_col() routine is given in the listing of Figure 7, and its purpose is to ensure that the first pixel in the data row starts in an even column number. The location of the first pixel in the row flags the start of the data packets, which will be further described below.

Line 12 begins a loop to examine the runs of pixels in the data row. Runs between the defined values MINRUN and MAXRUN are examined by the loop. The j-loop, and the test at line 15, locate a run of pixels, and sets the variable, k, to the index of the start of the run. The test at line 21 selects only blocks of pixels having length, i, less than the length of the row. The loop in line 22 moves the pixel run to temporary storage in the array block[].

The two tests at lines 24 and 25 reject blocks having run lengths other than the one required by the current value of the i-loop. The embedding scheme selects blocks of length, i, for embedding by adding a pixel to make the length i+1. This assures that the run can contain either i or i+1 non-zero pixel values, according to the bit-stream of the auxiliary embedded data. If the run stored in the variable block[] array does not end in at least two zeroes, it is not acceptable as a run of length, i+1, and the code branches to NEXT, to examine the next run found.

Line 28 begins a loop to count the number of pixels in the run. The number found is incremented by one in line 31 to account for the pixel added to make the run length equal to i+1. Line 33 contains a test ensuring that the run selected has the correct length. The histogram[] array for the run-length index, i, is incremented to tally the occurrence frequency of the run. The data row bytes for the run are flagged by the loop in line 36, with a letter code used to distinguish the runs located. This flagging technique permits the embedding code to identify easily the runs to be used for embedding the bit-stream. On exit from

this routine, the data row bytes contain runs flagged with letter codes to indicate the usable pixel positions for embedding the bit-stream. The return value is the number of runs located in the data row. A return of zero indicates no runs within the defined limits of MINRUN and MAXRUN were located.

FAX modem protocols emphasize speed, and therefore do not include error-correction. For this reason, FAX transmissions are subject to drop-outs, to impulsive noise, and to lost data, depending on the quality of the telephone line and the speed of the transmission. For successful embedding, the present invention must account for the possible loss of some portion of the image data. To accomplish this, a variation of modem block-protocols is used to embed the header and the auxiliary data. The two color image is treated as a transmission medium, with the data embedded in blocks, or packets, providing for packet-start flags, and parity checks. The start of a packet is signaled by an image row having its first pixel in an even column. The packet ends when the number of bits contained in the block are extracted, or, in the case of a corrupted packet, when a packet-start flag is located in a line. A checksum for parity, and a packet sequence number, are embedded with the data in a packet. Using this method, errors in the FAX transmission result in possible loss of some, but not all, of the embedded data.

The amount of data lost because of transmission errors depends on the density of pixels in the source image and the length of a dropout. Using 20 bytes per packet, a large

dropout in transmission of standard text results in one or two packets of lost data. Generally, the success of the invention follows the legibility of the faxed host image information.

Turning now to Figure 7, there can be seen a listing of the steps necessary to initialize the two color BITMAP® lines to flag the start of each packet. Each row in the two color image contains a non-zero value beginning in an even column (packet start), or in an odd column (packet continuation).

In Figure 7, it can be seen that line 4 starts a loop over the number of pixels in a data row. In FAX images, a zero (0) pixel value indicates a black space, and a one (1) value indicates a white space. Line 5 locates the first black space in the data for the row. If the variable, packet_size, is positive, the column index is tested to be even and the pixel is forced to be a white space. If the packet_size variable is negative, the routine returns an indicator of the data row flag without making changes. If packet_size is greater than zero, the first data row element is flagged as a white space. Line 11 deals with the case in which packet_size=0, indicating a continuation row. In the event of a continuation row, the first data row element is forced to a black space. The values returned by subroutines in lines 17-20 show the nature of the pixel row examined.

The code fragment listed in Figure 8 provides auxiliary data embedding into two color BITMAP® FAX images. The pixels in a row are processed as described above by examining the contents of the data row after it has been

analyzed and flagged with letter codes to indicate the run lengths. Lines 1 through 49 are part of a large loop (not shown) over the pixel index, lj, in the two color BITMAP® image. Lines 1-26 handle the reading of one line of pixels from the two color BITMAP®, and store the row number of the image in the variable, nrow, in line 1. The pixel value bits are decoded and expanded into the image_row[] array in lines 12-36. The image_row[] array contains the pixel values stored as one value (0 or 1) per byte.

Line 28 uses the packet_col() routine to return the packet-index for the row. If j is 0 in line 28, the row is a packet-start row, and if j is 1, the row is a continuation row. Line 29 uses the rowstats() routine to assign run-length letter flags to the pixels in the row buffer. The return value, i, gives the number of runs located in the image row. Consistency tests are made at lines 31, 37, and 41. The index, kp, gives the pixel row number within a data packet. If kp is 0, the line must be a packet-start index, and if kp>0, the line must be a continuation line. Line 49 completes the process of reading and preprocessing a row of two color image data.

The data-structure array, pair[], contains the run length for (i), the augmented run length, (i+1), and the total number of runs in the two color BITMAP® image. The index, k, in the loop starting at line 51, is the index for the run lengths being embedded. The index, inrow, counts pixels within the image row buffer, and the variable, bitindex is the bit-position index in the bit-stream byte.

Line 57 sets the value of the run-length letter-code in the variable, testltr. The value of an image pixel is tested against the letter-code in line 58. If the test letter-code flag is located, line 60 advances the index in the row to the end of the pixel run being used for embedding. The test function in line 62 checks the value for the current bit index in the bit-stream packet byte. If the value is one, the last pixel in the run is set to one. Otherwise, the last pixel in the run is set to 0.

Setting the value of the pixel trailing a run implements the embedding in the two color BITMAP® images by introducing noise generated according to the pseudo-random bit-stream in the packet data. The letter flag values written into the row buffer by the call to rowstats() in Figure 8 are reset to binary unit value before the image_row array data are packed and written back to the .BMP format file. The process for doing this is not illustrated in Figure 8, but is straightforward for those skilled in the art.

Extraction of data embedded into a two color BITMAP® FAX image, according to the present invention, can be accomplished only if the transmission of the FAX is received by a computer. The image data are stored by the receiving computer in a file format (preferably a FAX compressed format), permitting the processing necessary to convert the image to BITMAP® format and to extract the embedded data. FAX data sent to a standard office machine are not amenable to data extraction because the printed image is generally

not of sufficient quality to allow for recovery of the embedded data through scanning.

However, the invention does apply to scanning/printing FAX machines that process data internally with computer hardware. Auxiliary embedded data are inserted after the scanning of the host data, but prior to transmission. The auxiliary embedded data are extracted after they have been received, but before they are printed.

The key for two color image embedding can be recovered by analyzing the embedded image, because the run lengths are not changed from the original (i,i+1) values. The order in which the values are used depends on the frequency of occurrence in the image. As in the example for palette-color images, a key to the value and order of the pairs used for embedding is inserted into the FAX. However, the key is not strictly required, because, in principle, knowledge of the defined values MINRUN and MAXRUN permits re-calculating the run-length statistics from the received image. In practice, the key is required because transmission errors in the FAX-modem communication link can introduce new run-lengths that alter the statistical properties of the image, and because the pair ordering is not known. Even though FAX embedding is somewhat less secure than embedding auxiliary data into palette-color images, the two color BITMAP® FAX embedding of data still can be regarded as similar to one-time-pad cryptography.

The foregoing description of the preferred embodiments of the invention have been presented for purposes of illustration and description. It is not intended to be

exhaustive or to limit the invention to the precise form disclosed, and obviously many modifications and variations are possible in light of the above teaching. The embodiments were chosen and described in order to best explain the principles of the invention and its practical application to thereby enable others skilled in the art to best utilize the invention in various embodiments and with various modifications as are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the claims appended hereto.

WHAT IS CLAIMED IS:

1. A method of embedding auxiliary data into host data comprising the steps of:

creating a digital representation of said host data in the form of elements having numerical values and containing a noise component;

creating a digital representation of said auxiliary data in the form of a sequence of INDIVIDUAL bit VALUES;

evaluating said noise component of said digital representation of said host data;

comparing pairs of said elements with said noise component to determine pairs of said elements having numerical values which differ by less than said value of said noise component;

replacing individual values of said elements with substantially equivalent values from said pairs of elements in order to embed individual bit values of said auxiliary data corresponding to said sequence of bit values of said auxiliary data; and

outputting said host data with said auxiliary data embedded into said host data as a file.

2. The method as described in Claim 1 further comprising the step of combining said auxiliary data with predetermined information indicative of said auxiliary data, its file name, and file size, said step to be performed after the step of digitizing said auxiliary data.

3. The method as described in Claim 1 further comprising the step of determining a protocol for embedding said auxiliary data into said host data which allows for verification of said auxiliary data upon extraction from said host data.

4. A method of extracting embedded auxiliary data from host data containing a noise component comprising the steps of:

extracting from said host data a bit sequence indicative of said embedded auxiliary data, and which allows for verification of said host data;

interpreting said host data to determine host element pairs which differ by less than said noise component and which correspond to bit values of said auxiliary data;

identifying said auxiliary data using said bit sequence; and

extracting said auxiliary data as a file.

5. The method as described in Claim 1, wherein said host data comprises a color photograph.

6. The method as described in Claim 1, wherein said host data comprises a black and white photograph

7. The method as described in claim 1, wherein said host data comprises a television signal.

8. The method as described in Claim 1, wherein said host data comprises a painting.

9. The method as described in Claim 1, wherein said host data comprises a facsimile transmission.

10. The method as described in Claim 1, wherein said host data comprises an identification card.

11. The method as described in Claim 1, wherein said host data comprises digital audio information.

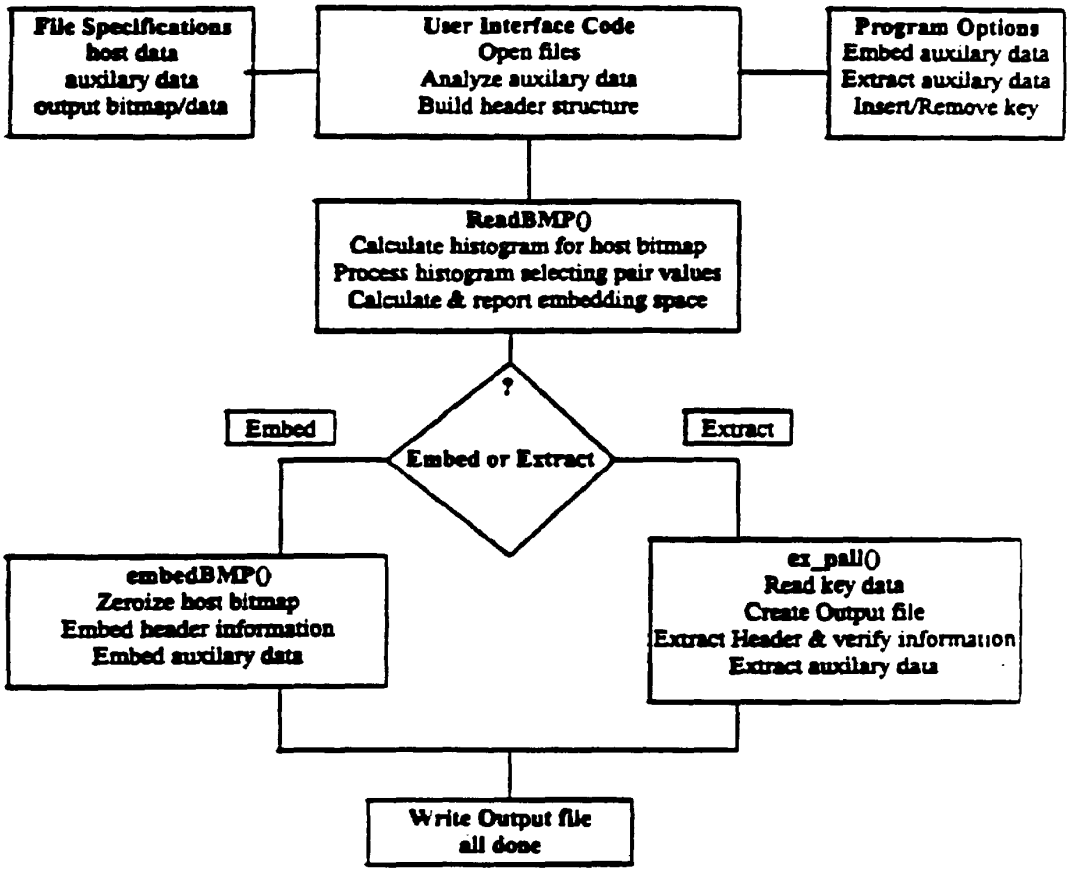

File Specifications
host data
auxilary data
output bitmap/data
User Interface Code
Open files
Analyze auxilary data
Build header structure
Program Options
Embed auxilary data
Extract auxilary data
Insert/Remove key
ReadBMP()
Calculate histogram for host bitmap
Process histogram selecting pair values
Calculate & report embedding space
?
Embed or Extract
Embed
Extract
embedBMP()
Zeroize host bitmap
Embed header information
Embed auxilary data
ex_pall()
Read key data
Create Output file
Extract Header & verify information
Extract auxilary data
Write Output file
all done

Fig. 1/8

Computer code for determining pair values.

```
line 1      for(i = 0; i < (int)bh.colors; i++) {
              int avg;
              if(i % 10 == 0)fprintf(stderr,".");
              c1.red = colormap[i].r;
              c1.grn = colormap[i].g;
              c1.blu = colormap[i].b;
line 7                  if(greyscale) {
                        avg = (int)(c1.red + c1.grn + c1.blu)/3;
                        if(avg==0)continue;
                        if(avg!=c1.red || avg!=c1.grn ||
                        avg!=c1.blu)continue;
                        }
line 12       (void)rgbhsi(&c1, &d1);     /* convert to HSI
                components */
              if((int)d1.inten == 255)unused++;
              old_diff = 0.f;
              if((int)d1.inten==0 || (int)d1.inten==(int)bh.colors)continue;
line 16     for(j=i+1; j < (int)bh.colors; j++)    {
                c2.red = colormap[j].r;
                c2.grn = colormap[j].g;
                c2.blu = colormap[j].b;
line 20         (void)rgbhsi(&c2, &d2); /* convert to HSI
                      components */
                color_diff = d2.hue - d1.hue;
```

```
                /* hue & inten. difference must be ok */
                if(!greyscale)      {
line 24         if((abs((int)color_diff) < COLOR_NOISE)\
                    && (color_diff < old_diff)\
                    && ((int)fabs((double)(d2.inten-d1.inten)) <
                       INTEN_NOISE) ) {
                        if(k>(int)bh.cols/2 -1)break;
                        pair[k].i = i;
                        pair[k].j = j;
                        pair[k].count = 0;
                        k++;
                        old_diff = color_diff;
                        }
                 }
line 35         else {
                    avg = (int)(c2.red + c2.grn + c2.blu)/3;
                    if(avg==0)continue;
                    if(avg!=c2.red || avg!=c2.grn || avg!=c2.blu)continue;
                    if( (int)fabs((double)(d2.inten-d1.inten))<
                       INTEN_NOISE &&
                          (int)fabs((double)(d2.inten-d1.inten)) !=0)   {
                             if(k>(int)bh.cols/2 -1)break;
                             pair[k].i = i;
                             pair[k].j = j;
```

Fig. 2/8

```
                pair[k].count = 0;
                k++;
                }
            }
        } /* j loop */
    if(k>(int)bh.cols/2 -1)break;
    } /* i loop */
line 51     no_pairs = k;
```

Fig. 2/8

Computer code to eliminate duplicate pairs.

```
line 1          for(i=0;i<k;i++) {
                pair[i].count += (hist_values[pair[i].i] +
                      hist_values[pair[i].j]);
                if(pair[i].i==0 || pair[i].j==0)pair[i].count = 0;
                }
line 5          p_sort(pair, k);
          no_pairs = duplicate (k, pair);
          total = 0;
          for(i=0;i<no_pairs;i++)
             total += pair[i].count;
line 10         total /= 8;
          value = (float)total - (float)NCOLS;
          if(value > 0.f) fprintf(stderr,"\n%.1f Kb embedding space
                located", value/1000.f);
          if(value == 0.f)fprintf(stderr,"\nNo embedding space available in
                this image");
          if(value < 0.f) fprintf(stderr,"\nInsufficient embedding space");
```

Fig. 3

Computer code constraining pair values for Truecolor images.

```
line 1        /* Find histogram point-pairs within RANGE counts, within
                 10% in number */
          for(ip=0;ip<3;ip++)  {
              int nstart;
              long li;
              fprintf(stderr,"\nAnalyzing intensity histogram for plane %d", ip);
line 6              for (i=0;i<256;i++) {
                   fvalue[i]=(float)hist_values[ip*256+i];
                   }
              nstart = RANGE;
              k = 0;
              while(nstart<256 && k<(int)bh.cols/2)    {
line 12                     for (i=nstart;i<256;i++) {
                            for(j=i-1;j>i-RANGE;j--) {
                                  li = hist_values[ip*256+i];
                                  if((int)(fvalue[j]*fvalue[i])==0)break;
                                  if((float)fabs((double)(fvalue[j]-fvalue[i]))\
line 17                                       < 0.05f*(fvalue[j]+fvalue[i])) {
                                 pair[k].i = i;
                                 pair[k].j = j;
                                 pair[k].count = li + hist_values[ip*256+j];
                                       k++;
```

Fig. 4/8

```
                    if(k>(int)bh.cols/2-1)break;
                    }
                    } /* end of inner pixel comparison loop
                         (j) */
               if(k>(int)bh.cols/2-1)break;
               } /* end of outer pixel comparison loop (i) */
line 27        p_sort(pair, k);
          no_pairs = duplicate (k, pair);
          k = no_pairs;
              if(verbose)fprintf(stderr,"%3d pairs\r", k);
line 31       nstart = i;
          } /* end of while loop */
```

Fig. 4/8

Computer code for embedding auxiliary data.

```
/*----------EMBEDDING CODE----------*/
/*
        lj,     index over pixels in the image-data
        inrow,      index within the image-data row buffer
        nrow, row number in the image-data
        li,     index over pixels in the data-image
        d_inrow, "  within the data-image row buffer
        k,      index within the PAIRS structure array
        maxval,  no. of bits embedded
        bitindex, bit position within the data-image byte
        byteplace,  position for read/write in tape6 file
*/
line 1      data_row = (unsigned char *)malloc((size_t)NCOLS);
            if(data_row==NULL) {
            pm_error("Data row data allocation failed!");
            return(1);
            }
            maxval = bit_place_index.maxval;
            d_inrow = bit_place_index.d_inrow;
            bit_place_index.li += d_inrow;
            lj = (long)krow;
            k  = 0;
            nrow = -1;
line 12     for(li=bit_place_index.li; li<length-NCOLS; li++)      {
            bit_place_index.li = li;
line 14         if(li == -512L)     {       /* header information */
                byteptr=(unsigned char *)&data_header;
                for(d_inrow=0;d_inrow<sizeof(data_header);d_inrow++)
                                data_row[d_inrow]=*(byteptr+d_inrow);
                d_inrow = 0;
                }
```

Fig. 5/8

SUBSTITUTE SHEET (RULE 26)

```
line 20     if((li >= 0L && (li % (long)NCOLS) == 0L) || reread != 0)
                { /* next row of data-image */
            j = fgetpos(tape5,&tape5_pos);
            j = fread(data_row, 1, (size_t)NCOLS, tape5);
            if(!reread) {
                    for(i=0;i<j;i++) checksum += data_row[i];
                    d_inrow = 0;
                    bit_place_index.d_inrow = 0;
                    }
            reread = 0; /* turn off flag for re-read on next Truecolor
                    plane */
            }
line 30     for (bitindex=bit_place_index.bitplace;
                    bitindex<NO_BITS; bitindex++) {
            bit_place_index.bitplace = bitindex;
line 32             if((lj-krow) % (long)(BYTES_IN_ROW) == 0L) {
                    if(nrow >= 0) {         /* write only after you read */
                            inrow = fseek(tape6, byteplace, SEEK_SET);
                            inrow =fwrite(image_row,1,(size_t)
                                    (BYTES_IN_ROW),tape6);
                            byteplace += inrow;
                            byteplace += pad;  /* skip pad bytes */
                            inrow = krow;
                            }
line 40     if(lj/(long)OFFSET == 0L ||
            (lj+(BYTES_IN_ROW+pad))/
            (BYTES_IN_ROW+pad) > (unsigned \long)bh.rows)
                            {
                            if(bailout()) { /* end of
                                    image-data--user
                                    termination */
                               i = 1;
```

Fig. 5/8

```
                    goto QUIT;
                    }
line 47             if(k==no_pairs)goto PLANE;  /* next plane of
                        image */
                    lj = krow;    /* pick next pair and start over */
                    pvalue.i=(unsigned int)pair[k].i;  /*zero*/
                    pvalue.j =(unsigned int)pair[k].j; /* one */
                    if(verbose && k>0) fprintf(stderr," %ld     ",
                        pvalue.count);
                    pvalue.count = 0L;
                    if(verbose) fprintf(stderr,"\rEmbedding Pair
                        %2d\ (%3d,%3d)",\
                        k, pvalue.i, pvalue.j);
                    else fprintf(stderr,".");
                    k++;
                    byteplace = bh.pixeloffset;
                    }
line 60         nrow = (int)((lj-krow)/((long)BYTES_IN_ROW+pad));  /*
                    read next row */
                inrow = fseek(tape6, byteplace, SEEK_SET);
                inrow = fread(image_row,1,(size_t)
                    BYTES_IN_ROW,tape6);
                inrow = krow;
                } /* end new row (lj) test */
            /* Embed one byte */
             if(ip>=0 && pair[k-1].count==0)  {  /* finished a pair */
                lj +=OFFSET;
                inrow += OFFSET;
                bitindex--;
                continue;
                }
```

Fig. 5/8

```
line 72          if((int)image_row[inrow]==pvalue.i) {     /* find a zero
                        value */
                        if(test((int)data_row[d_inrow],bitindex))
                                image_row[inrow]=(unsigned char)pvalue.j;
                                maxval++;
                        if(pair[k-1].count==0)  {
                                pm_error("\nPair count error!");
                                i = 1;
                                goto QUIT;
                                }
                        pair[k-1].count--;
                        pvalue.count++;
                        if(bitindex==NO_BITS-1)bit_place_index.bitplace
                                =0;
                        }
line 84          else bitindex--;    /* haven't got this bit yet! */
                 lj+=OFFSET;
                 inrow+= OFFSET;
                 } /* end of bitindex loop */
        d_inrow++;
        } /* end of li (data index) loop */
```

Fig. 5/8

Computer code to analyze lengths of runs.

```
line 1      int rowstats(unsigned char *data_row, long *histogram, int
                ncols, int packet_size) {
              int i, j, k, l;          /* loop counters */
              int runs=0;              /* return value */
              int count;               /* no. of pixels in the run */
              char letter = 'A';       /* starting code for flagging runs
                                          in the row */
              unsigned char block[MAXRUN+3];  /* a block
                                          containing the run being
                                          examined */
            /* find first bit in the row & adjust as a packet flag */
            if(packet_size >=0) {
line 9      j = packet_col(data_row, packet_size, ncols);
            }
            if(ncols <=0) return(-1);
line 12     for(i=MINRUN;i<=MAXRUN;i+=2) { /* i is the runlength being
                searched */
              k = 0;
              for(j=1;j<ncols;j++) { /* NOTE: data_row[0] is assumed
                to be zero!! */
line 15         if(data_row[j]==(unsigned char)ONE) {
                  if(data_row[j-1]!=(unsigned char)ZERO) continue;
```

Fig. 6/8

```
                    k = j;  /* a block start */
                }
              else continue;
        /* find a block of data ending with a zero pixel */
line 21       if(k+i+2 > ncols) break;
line 22       for(l=k;l<k+i+3;l++) block[l-k] = data_row[l];
              l = j;
line 24       if(block[i+1] != (unsigned char)ZERO) goto NEXT;
              if(block[i+2] >  (unsigned char)ONE ) goto NEXT;
        /* examine block for pixel count */
        count = 0;
line 28 for(l=0;l<i;l++) { /* all but last bit in block must = 1 */
            if(block[l]==(unsigned char)ONE) count++;
            }
line 31  count++;
         l = j+1;
line 33  if(count == i+1) { /*  set all but last pixel in run to flag value */
            if(histogram != NULL)histogram[i]++;
                                          runs++;
line 36     for(l=j;l<j+count-1;l++) data_row[l] = letter;
            l++
            }
            NEXT:  for(j=l+1;j<NCOLS;j++) if(data_row[j]==(unsigned
                   char)ZERO)break;
```

Fig. 6/8

```
        } /* end of row (j) loop */
        letter++;
        } /* end of run (I) loop */

return(runs);
}
```

Fig. 6/8

Computer code to set packet-start pixel flag.

```
int packet_col(unsigned char *data_row, int packet_size, int
      ncols) { int i;
/* find first bit in the row & adjust as a packet flag */
line 4   for(i=1;i<ncols;i++)     {
line 5         if(data_row[i]==(unsigned char)ZERO)  {
                 if(packet_size<0) break;
                 if(packet_size>0)    {      /* first bit set to an even
                                              column */
                    if(i%2 == 0)break;
                    data_row[i] = (unsigned char)ONE;
                    }
line 11          else { /* first bit set to an odd column */
                    if(i%2 != 0)break;
                    data_row[i] = (unsigned char)ONE;
                    }
                 }
               }
line 17  if(i==ncols)return(-1);  /* no black pixels in the row */
         if(packet_size>=0) return(i); /* index of the first black pixel */
         if(i%2) return(1);          /* if(packet_size==-1) return odd */
         else  return(0);            /*                    return even */
         }
```

Fig. 7/8

Computer code to embed data in two-color images.

```
line 1      READLINE:    nrow = (int)(lj/((long)bh.cols));  /* read
                                    data from next row */
            if(verbose) {
                if(nrow==0)fprintf(stderr,"\n");
                 fprintf(stderr,"\rrow %4d", nrow);
                 }
            else motion(stderr);
            bit_count = 0;
            image_row[0] = 0;       /* row buffer always starts
                                        with a zero */
            if(verbose==2 && nrow <=61)fprintf(tape9,"\nnrow
                 byteplace %d %ld",     nrow,byteplace);
            inrow = fseek(tape6, byteplace, SEEK_SET);
            writeplace = byteplace;
line 12     for(j = 1; j < (int)bh.cols+1; j++) {
                int pix;
                if(bit_count <= 0) {   /* need another byte */
                    bit_count = 8;
                    bit_store = pbm_getrawbyte(tape6);
                    byteplace++;
                    }
```

Fig. 8/8

```
                    bit_count -= bh.bitsperpixel;
                    pix = ( bit_store >> bit_count ) & mask;
                    image_row[j] = (unsigned char)pix;
                    #ifdef INSERT_KEY
                    /* key row set to zero to hold key pairs */
                    if(nrow == KEYLINE)image_row[j] = (unsigned
                        char)ZERO;
                    #endif
line 26             } /* cols */
                byteplace += pad;
line 28         j = packet_col(image_row,-1,(int)bh.cols);
                i = rowstats(image_row,NULL,(int)bh.cols+1,-1);   /*
                    flag the embedding pixels */
                if(verbose==2) fprintf(tape9,"\n nrow,i,j: %d %d
                    %d", nrow,i,j);
line 31         if(j<0 || i==0)      { /* a row of white pixels or no
                        pixels for embedding   */
                    if(nrow+1<(int)bh.rows) {
                        lj += bh.cols;
                        goto READLINE;
                        }

                    }
line 37         if(j==1 && kp==0) {
```

Fig. 8/8

```
                        fprintf(stderr,"\nPacket start-index error,
                              packet %d", packet_no-1);
                        goto QUIT;
                        }
line 41                 if(j==0 && kp > 0) {
                        fprintf(stderr,"\nContinuation packet-index
                              error, packet %d", packet_no-1);
                        goto QUIT;
                        }
                        inrow = 1;
                        if(kp==0 && verbose==2)
                          fprintf(tape9,"\nPacket start-row %d,  bits found
                            %d",nrow,i);
                        kp++;
line 49                 } /* end new row (lj) test */
                /* Embed one byte, use all pairs for each row */
line 51         for(k=0;k<no_pairs;k++) {
                    if(pair[k].count<0) {
                        pm_error("\nPair count error!");
                        i = 1;
                        goto QUIT;
                        }
line 57             testltr = (unsigned char)(letter+(unsigned char)pair[k].i/2 -
                              1);    /* flag letter */
```

Fig. 8/8

```
if(image_row[inrow]==testItr)      {          /* find a flagged run
        */
    if(verbose==2 && nrow==60) fprintf(tape9,"inrow %d",
        inrow);
line 60    inrow += (unsigned int)pair[k].j;
    lj += pair[k].j;
line 62    if(test((int)packet[inpacket_row],bitindex)) image_row
        [inrow-1]=1;
    else image_row[inrow-1]=0;
```

Fig. 8/8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US96/02357

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :G06F 17/30
US CL : 395/600;380/4,49,55
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 395/600;380/4,49,55

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS, DIALOG (FILES 2,6,8,35,347,348,350,351,256,275), PROQUEST, DR-LINK,INTERNET

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US,A 4,965,827 (McDonald) 23 October 1990, vol 2, lines 35-56	1-2
A	US,A 4,972,476 (Nathans) 20 November 1990 Abstract, Disclosure	1-2
A	US,A 5,291,243 (Heckman et al.) 1 March 1994 Abstract, Disclosure	1-2
A	US,A 5,321,749 (Virga) 14 June 1994 Abstract Disclosure	1-2
A,T	US,A 5,398,283 (Virga) 14 March 1995 Abstract, Disclosure	1-2
A,T	US, A 5,488,664 (Shamir) 30 January 1996	1-2

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be part of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search	Date of mailing of the international search report
19 JUNE 1996	12 JUL 1996
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	Authorized officer PAUL R. LINTZ
Facsimile No. (703) 305-3230	Telephone No. (703) 305-9600

Form PCT/ISA/210 (second sheet)(July 1992)*

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US96/02357

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A,T	US, A 5,493,677 (Balogh et al.) 20 February 1996 Abstract, Disclosure	1-2
A	Proceedings of the Second International Conference on Document Analysis and Recognition, Tsukuba City, Japan, 20-22 October 1993, pp 95-98, Brzakovic et al., "Document Recognition/Authentication Based on Medium-Embedded Random Patterns", pp. 95-98	1-2
A	Proceedings ICIP-94, Austin Texas, 13-16 November 1994, pp. 86-90, Van Schyndel et al., pp. 86-90	1-2
A,T	Fifth International Conference on Image Processing and Its Applications, Edinburgh, UK, 4-6 July 1995, pp. 326-330,Boland et al., "Watermarking Digital Images For Copyright Protection", page 326-330	1-2
A,T	Dr. Dobbs Journal, #229, April 1995, pp. 18-26, 82-87, Steve Walton, "Image Authentication For a Slippery New Age", page 18-26.	1-2

Form PCT/ISA/210 (continuation of second sheet)(July 1992)*

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06K — A2

(11) International Publication Number: WO 99/36876

(43) International Publication Date: 22 July 1999 (22.07.99)

(21) International Application Number: PCT/US99/01296

(22) International Filing Date: 20 January 1999 (20.01.99)

(30) Priority Data:
60/071,983 20 January 1998 (20.01.98) US

(71) Applicant: DIGIMARC CORPORATION [US/US]; Suite 500, One Centerpoint Drive, Lake Oswego, OR 97035 (US).

(72) Inventors: RHOADS, Geoffrey; 304 S.W. Tualatin Loop, West Linn, OR 97068 (US). GUSTAFSON, Ammon; 13305 Lacewood Street, Beaverton, OR 97008 (US).

(74) Agent: GALBI, Elmer; 13314 Vermeer Drive, Lake Oswego, OR 97035 (US).

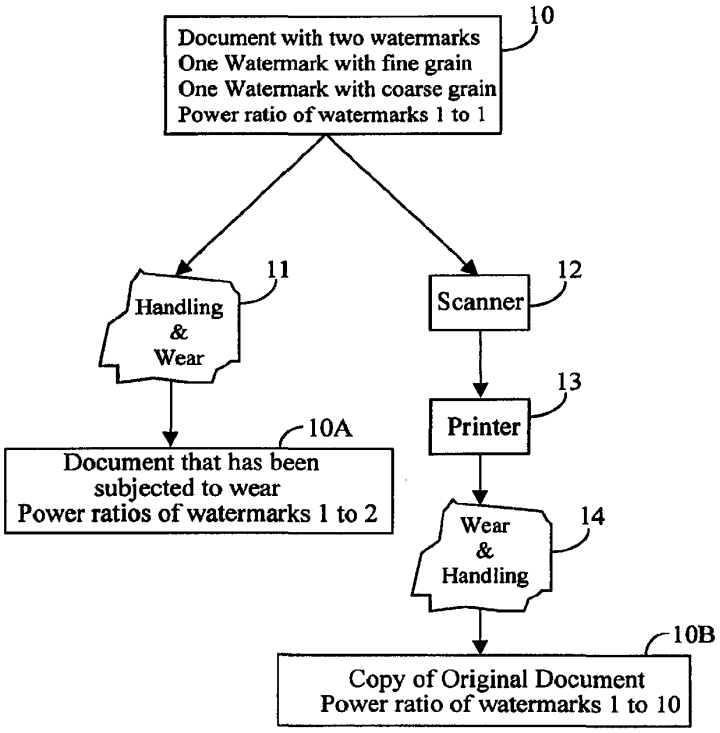
(81) Designated States: AU, BR, CA, IL, JP, KR, MX, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: MULTIPLE WATERMARKING TECHNIQUES

(57) Abstract

Multiple digital watermarks, each of which has different characteristics, are embedded in a document. The characteristics of the various watermarks are chosen so that each of the watermarks will be affected in a different manner if the document is subsequently copied and reproduced. The detection process or mechanism reads each of the watermarks and compares their characteristics. While wear and handling may change the characteristics of the digital watermarks in a document, the relationship between the characteristic of multiple digital watermarks in a document will nevertheless give an indication as to whether a document is an original or a copy of an original.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

MULTIPLE WATERMARKING TECHNIQUES

Field of the Invention:

The present invention relates to steganography and more particularly to the use of watermarks to determine the authenticity and history of a particular document or image.

Background of the Invention:

Steganographic and digital watermarking technologies are well know. For example see U.S. Patent 5,636,292 and the extensive references cited therein. Also see copending patent applications serial number 08/327,426 which was filed 10/21/94 and copending application serial number 08/436,134 which was filed 5/8/95.

The technology for inserting digital watermarks in images and the technology for reading or detecting digital watermarks in images is well developed, well known and described in detail in public literature. Furthermore, there are commercially available products which include programs or mechanisms for inserting digital watermarks into images. For example the commercially available and widely used products "Adobe Photoshop" which is marketed by Adobe Corporation of San Jose California and "Corel Draw" program which is marked by Corel Corporation of Ontario Canada, include a facility for inserting digital watermarks into images.

The technology for making high quality copies of documents is widely available. The technical quality of scanners and color printers has been increasing rapidly. Today for a relatively low cost one can purchase a high quality scanner and a high quality color printer. Thus, it is becoming increasingly easy to duplicate documents. The ability to create high quality copies has created a need for technology which can differentiate between original documents and copies of the original.

It is known that watermarks can be used to help differentiate genuine documents from copies. However, the prior art techniques for using digital watermarks to differentiate genuine documents from copies have serious limitations. The present invention is directed to an improved technique for using steganography and digital watermark technology to facilitate differentiating original documents from copies of the original.

The present invention can also be used for various other purposes such as to embed multiple types of information in a single document or to provide watermarks which enable documents to perform special functions. .

Summary of the Invention:

With the present invention multiple digital watermarks, each of which has different characteristics are embedded in a document. The characteristics of the two watermarks are chosen so that each of the watermarks will be affected in a different manner by what may subsequently happen to the document.

The detection process or mechanism reads the two digital watermarks and compares their characteristics. While wear and handling may change the characteristics of the individual watermarks, the relationship between the characteristic of the two watermarks will never-the-less give an indication as to whether a document is an original or a copy of an original.

For example according to the present invention two digital watermarks in a document may have different energy levels. The absolute energy level of a digital watermark in an original image may be decreased if a document is subject to wear.

Likewise the energy level of the digital watermark in an image may be decreased if an image is scanned and reprinted on a color printer. However, the relationship between the energy level of the two digital watermarks will be different in an image that has been subject to wear and in a reproduced image. Likewise if two digital watermarks are introduced into an image where the bit pattern used to construct the digital watermarks have different patterns, the ratio between the signal to noise ratio of the watermarks will be different in an original subject to wear and in a copy generated by scanning the original and printing the scanned image. Other characteristics of multiple digital watermarks can also be used to differentiate original documents from copies.

Brief Description of the Figures:

Figure 1 shows the paths that a document and a copy may follow.

Figures 2A and 2B show a fine grain and a course grain watermark.

Figure 3A and 3B show a geometrically linear and a geometrically random assignment of pixels to a bit in a digital watermark.

Figure 4 illustrates a fourth embodiment of the invention.

Detailed Description of preferred embodiments:

The problem of differentiating an original document from a copy is made more difficult in situations where the original document is subject to being handled, worn, folded and otherwise damaged. Many original documents such as identification documents and currency are extensively handled. The wear to which such documents are subjected reduces the quality of images on the document and therefore reduces the quality of any information embedded in the document using conventional steganographic techniques.

With the present invention a number of different watermarks are embedded in a document. Each of the watermarks embedded in the document, has different characteristics. All watermarks are somewhat affected when a document is subjected to wear, and all watermarks are somewhat affected when a document is duplicated by being scanned and reprinted. However, the magnitude of the effect caused by being scanned and reprinted on watermarks with certain characteristics is much greater than the effect on watermarks with different characteristics. Likewise, wear and handling of a document affects watermarks with certain characteristics much more than it affects watermarks with different characteristics.

Thus, if multiple watermarks with different characteristics are inserted into a document, it is possible to differentiate a copy from an original document that has been subjected to wear by examining the ratios of characteristics of the watermarks in the image being examined.

In order to print a document on a color printer, the document is put through a transformation from a color space such as the RGB color space to a different color space such as the CMYK (cyan, magenta, yellow, black) color space. Such transformations are well know. For example see chapter 3 entitled "Color Spaces" in a book entitled "Video Demystified, A handbook for the Digital Engineer", Second Edition, by Keith Jack, published by Harris Semiconductor and Hightext Publications of San Diego, California.

When an image is transformed from one color space to another color space, noise is introduced into the image Among the reasons for this is the fact that each color space has its own distinctive gamut (or range) of colors. Where the gamut of two color spaces overlap, the conversion from one color space to another color space

can in theory be precise. However, there will be some areas which are in the gamut of one color space not in the gamut of another color space. Such situations definitely introduce noise into the conversion process. Even in areas that are in the gamut of two color spaces, conversion from one color space to another color space introduces noise because of such things as round off errors. The present invention takes advantage of the fact that if an original is copied and then a copy is printed, the image on the printed copy will have gone through several conversions to which the original will not have been subjected. For example, the conversions to which a copy may be subjected

are:

1) a document to RGB conversion (i.e. scanning the document into the computer),

2) a RGB to CMYK conversion,

3) a CMYK to copy conversion (i.e. printing the document).

Any characteristics of the two digital watermarks that will be affected differently by the additional conversion process to which copies are subjected can be used to differentiate copies from an original. Since the two watermarks with different characteristics are affected in a different manner by the additional conversion step, a comparison of the characteristics of the two watermarks in a document being examined will indicate if the document is an original (which has not gone through the additional conversions) or a copy which has gone through the additional conversions. While the characteristics of each watermark will have been changed by wear and by the copying process, the comparison between the characteristics of the two watermarks will still be able to differential a copy from an original.

Four embodiments of the invention are described below. Each of the embodiments utilizes two watermarks in a document. The differences between the two watermarks in the document are as follows:

In the first embodiment:

First watermark: Has fine grain

Second watermark: Has a course grain

In the second embodiment:

First watermark: Has geometrically linear assignment of pixels

Second watermark: Has geometrically random assignment of pixels.

In the third embodiment:

First watermark: Has low power

Second watermark: Has higher power

In the fourth embodiment:

Fist watermark: uses standard RGB to HSI and HSI to RGB transformations

Second watermark is biased before being transformed from HSI to RGB.

Figure 1 shows the steps to which documents and copies are typically subjected. In the normal course, a document 10 may be subjected to handling and wear 11 resulting in a worn document 10A. Document 10 may also be scanned as illustrated by box 12. The scanning produces a digital image which can be printed as illustrated by box 13. The printed image may be subjected to handling and wear 14 resulting in a copy 10B. It is noted that the document 10 may also be subject to handling and wear prior to the scanning operation 12. The task to which this invention is directed is the task of differentiating the worn document 10A from the copy 10B.

The document 10 includes an image (not explicitly shown) which has two digital watermarks inserted therein. In the first embodiment of the invention, the first watermark has a fine grain and the second watermark has a course grain. The grain of the two watermarks is illustrated in Figure 2. Figure 2A shows the grain of

the first watermark and figure 2B shows the grain of the second watermark. The first watermark uses blocks of 9 pixels (a 3 by 3 block). Each of the pixels in each 9 pixel block has its gray value changed by the same amount. For example Figure 2A shows that the first 9 pixel block has its gray value increase and the second 9 pixel block has its gray value decreased. The amount of increase and the selection of blocks that is increased and decreased is conventional.

As shown in Figure 2B, the grain of the second watermark is in blocks that are 6 pixels by 6 pixels or 36 pixels. All of the pixels in each 36 pixel block are changed by the same amount.

In the original document 10, the two watermarks have a power ratios of 1 to 1. After wear and handling, the power of the first watermark will be degraded somewhat more than the power of the second watermark. For example, as illustrated in Figure 1, after document 10 is subjected to handling and wear, a detector which reads the watermarks might find that the power ratio of the water marks is 1 to 2.

If the document 10 is scanned and the resulting digital image is printed to make a copy of the document 10, the ratio of the power of the watermarks will be affected much more than the effect of handling and wear. For example as illustrated in Figure 1, the power ratio of the watermarks may be 1 to 10, thereby allowing one to differentiate the worn original document 10A from the copy 10B.

It is noted that the mechanism for inserting watermarks into an image is well known as is the technique for reading a watermark and using correlation techniques to determine the signal to noise ratio (i.e. the power) of a watermark.

Figures 3A and 3B shown an alternative technique for implementing the present invention. In the second embodiment of the invention, the two watermarks inserted into the image on a document have different patterns of assigning pixels to the bits of the number which the watermark represents. The first watermark utilizes a geometrically linear assignment of pixels to each bit. For example Figure 3A shows an image that has 500 by 500 pixels. Considering a watermark with 50 bits, each bit of the watermark would have 5000 pixels assigned to represent that bit. A linear assignment could have each fifth bit in each row (100 bits per row) and each fifth row (50 rows) assigned to each bit of the watermark. Thus 5000 pixels would be assigned to each bit in a very orderly or linear manner.

In the second watermark the pixels would be assigned to each bit in a random manner as shown in Figure 3B. Each bit in the watermark would still have 5000 assigned bits; however, the pixels would be a random location over the image. Naturally it should be understood that Figure 3A and 3B illustrate how pixels are assigned to one bit of the watermark. The other bits of the watermarks would have pixels assigned in a similar manner.

Similar to the first embodiment of the invention, the watermark with a linear assignment of pixels and the watermark with a random assignment of pixels would be affected differently by handling and wear on the original document than they would be by being scanned and reprinted.

The third embodiment of the invention described herein utilizes watermarks which have different power levels. Handling and wear as contrasted to scanning and printing would affect a watermark with a low power level differently than a water mark with a high power level. Watermarks with different power levels can be

inserted into a document in order to practice the present invention utilizing commercially available programs such as Adobe Photoshop or Corel Draw. In the Adobe Photoshop and Corel Draw programs, the power or intensity of the watermark can be adjusted by setting a simple control setting in the program.

The fourth embodiment of the invention introduces different characteristics into two watermarks by modifications made to one of the watermarks during the initial step during which the watermarks are introduced into an image. The operation of the fourth embodiment can be explained as shown in Figure 4. First as illustrated by equation 1 there is a conversion from RGB to HSI as is conventional. This is illustrated by equation 1. As illustrated by equation 2, the first watermark is inserted into the image in a conventional manner by modifying the I value in the HSI representation of the image using the first watermark values (designated as WM1 Δ). A first RGB value designated RGB(1) is then calculated using a conventional transformation designated T. As indicated by equation 3, the second watermark WM2 is then biased toward a particular color and the biased watermark is then combined with the HSI values and transformed to a second set of RGB values designated RGB(2). Finally as indicated by equation 4, the values RGB(1) and RGB(2) are combined to form the watermarked image designated RGB(F).

The transform used to go from RGB to HSI color space (indicated in equation 1 in Figure 4) can be anyone of a variety of known other techniques. For example, the RGB to HSI conversion can be one of the techniques explained in the above referenced text book such as the following: (which assumes that RGB and Intensity have a value range of 0 to I and that Red equals 0°):

First calculate:

$$M = \max(R,G,B)$$

m = min (R,G,B)

r = (M-R)/(M-m)

g = (M-G) /M-n)

b = (M-B) / (M-m)

Then calculate I, S, and H as follows:

a) I = (M + M) /2

b) if M = m then S = 0 and H = 180

if I< or = 0.5 then S = (M-m)/(M+m)

if I > 0.5 then S = (M-m) / (2-M-m)

c) if R = M then H = 60 (b-g)

if G = M then H = 60 (2 +r - b)

if B = M then H = 60(4 + g - r)

if H > or = 360 then H = H · 360

if H < 0 then H = H + 360

The first watermark in inserted into the RGB values in a conventional manner by modifying the I value of appropriate pixels so as to combine the watermark Δ values with HSI values. This is indicated by equation 2 in Figure 4. Next as indicated by equation 3 in Figure 4, the HSI values are converted to RGB values using a transform "T". The transform "T" can be conventional and it can for example be done as follows:

First calculate:

if I < or = 0.5 then M = I (I + S)

if I > 0.5 then M = I + S - IS

m = 2I - M

if S = 0 then R = G = B = I and H = 180°

Then calculate R, G and B as follows:

a) if H < 60 then R = M

if H < 120 then R = m + ((M-m) / ((120- H) / 60))

if H < 240 then R = m

if H < 300 then R = m + ((M - m) / ((H - 240 / 60))

otherwise R = M

b) if H < 60 then G = m + ((M-m)/ (H/60))

if H < 180 then G = M

if H < 240 then G = m + ((M - m) / ((240 - H_ / 60))

otherwise G = m

c) if H < 120 then B = m

if H < 180 then B = m + ((M -m) / ((H-120/60))

if H < 300 then B = M

otherwise B = m + ((M - m) / ((360 - H) / 60))

Next the values which represent a second watermark are used to calculate a second set of RGB values designated RGB2. In order to calculate RGB2, the values of H and S are modified so that they are slightly biased toward a particular color designated H1 and S1 New values for H and S are calculated as follows: (Note, H1 must be between 0 and 360, S1 must be non-negative and can be between 0 and 1 and X is a value between 0 and 1)

Calculate new values for H and S as follows:

If H > H1 then H = H - (H - H1) x

else H = H + (H1 -H)x

If S > S1 then S= S - (S- S1)x

else S = S +(S1 -S) x

:

Next add the second watermark to the values of HSI and transform these values to the RGB color space as indicated by equation 3 The transformation from HSI color space to RGB color space is done as previously indicated.

Finally as indicated by equation 4 in Figure 4, the final RGB value (designated RGBF) is calculated by combining the values of RGB1 and RGB2. This combination can be done in a variety of known ways.

It is noted that in the above example the difference between the transformation used for the first and the second watermarks involves biasing the values of H and S. Alternatively a wide variety of different changes could also be made. The key to this fourth embodiment of the invention is that in effect a different transformation is used for the first and the second watermarks

While four embodiments of the invention have been shown herein , it should be understood that many other characteristics and attributes of a digital watermark could be used to practice the present invention in addition to the characteristics and attributes described herein. Furthermore other known digital watermarking techniques can be used together with and applied to the digital watermarks used for the present invention. It is also noted that while in the above examples only two watermarks were used, in some situations one could use three, four five or more watermarks. That is, the embodiments of the invention specifically described herein utilize two watermarks. It should be understood that any number of watermarks could be utilized in like manner. Furthermore while the embodiments shown herein utilize two separate watermarks, the two watermarks used to practice the present invention could be combined into one watermark which has a plurality of separate identifiable and measurable characteristics.

It is noted that while the present invention utilizes multiple watermarks with different characteristics to differentiate original documents from copies of the original, one can also utilizes multiple watermarks with different characteristics for other reasons. Documents may include multiple similar watermarks in addition to the watermarks which have different characteristics according to the present invention. As used herein, in general, the term "document" refers to a physical entity.

While the present invention has been described with respect to four specific embodiments of the invention, it should be understood that various changes in forma and detail could be made without departing from the spirit and scope of the invention. The scope of the present invention is limited only by the appended claims.

I claim

1) A document which has embedded therein a first digital watermark having a first set of characteristics and a second watermark having a set of characteristics which differ from said first set of characteristics.

2) The document recited in claim 1 wherein said first watermark has a different energy level from said second watermark.

3) The document recited in claim 1 wherein said first and second watermarks have bit patterns and wherein the bit patterns which comprise said first watermark are different from the bit patterns which comprise said second watermark

4) A method of creating a watermarked image which comprises the steps of:
modifying an HSI (hue, saturation, Intensity) representation of an image to imbed a first watermarked in said image and create a first watermarked image,
transforming said first watermarked image to the RGB (red, green, blue) color space,
biasing the values which represent a second watermark toward a particular color,
modifying a said HSI values to imbed said biased second watermarked in said image and create a second watermarked image,
transforming said second watermarked image to the RGB color space, and
combining the values first and second watermarked images to create a final watermarked image.

5) The method recited in claim 4 wherein said biasing is toward a particular color.

6) A method of differentiating copies of an original document from the original document comprising, said document containing a first digital watermark which has a first set of characteristics and a second digital watermark which has a second set of characteristics,
reading said first and second watermarks from said original document and comparing the resultant values to generate a first set of results,
reading said first and second watermarks from said copy of said original document and comparing the resulting values to generate a second set of results, and
using differences between said first and second sets of results to differentiate an original document from a copy of said original document.

7) The method recited in claim 6 wherein said first digital watermark has a first energy level and said second watermark has a second energy level.

8) The method recited in claim 6 wherein said first watermark has a first bit pattern and said second watermark has a different bit pattern.

9) The method recited in claim 6 where both said original document and said copy have been subjected to wear.

10) The document recited in claim 1 wherein said second watermark was been biased toward a particular color before it was inserted in said document.

11) A method of differentiating an original document from a copy of said document, said document having first and second watermarks imbedded therein, comprising the steps of comparing the characteristics of the said first and second watermarks in

said original document and comparing the characteristics of said first and second watermarks in said copy.

12) the method recited in claim 11 wherein said first and second watermarks have different grain structures.

13) The method recited in claim 11 wherein said first and second watermarks have different intensity levels.

14) The method recited in claim 12 wherein said first and second watermarks have different characteristics.

15) A document which contains an image which has embedded therein a plurality of watermarks, each of said watermarks having characteristics which differ from each other.

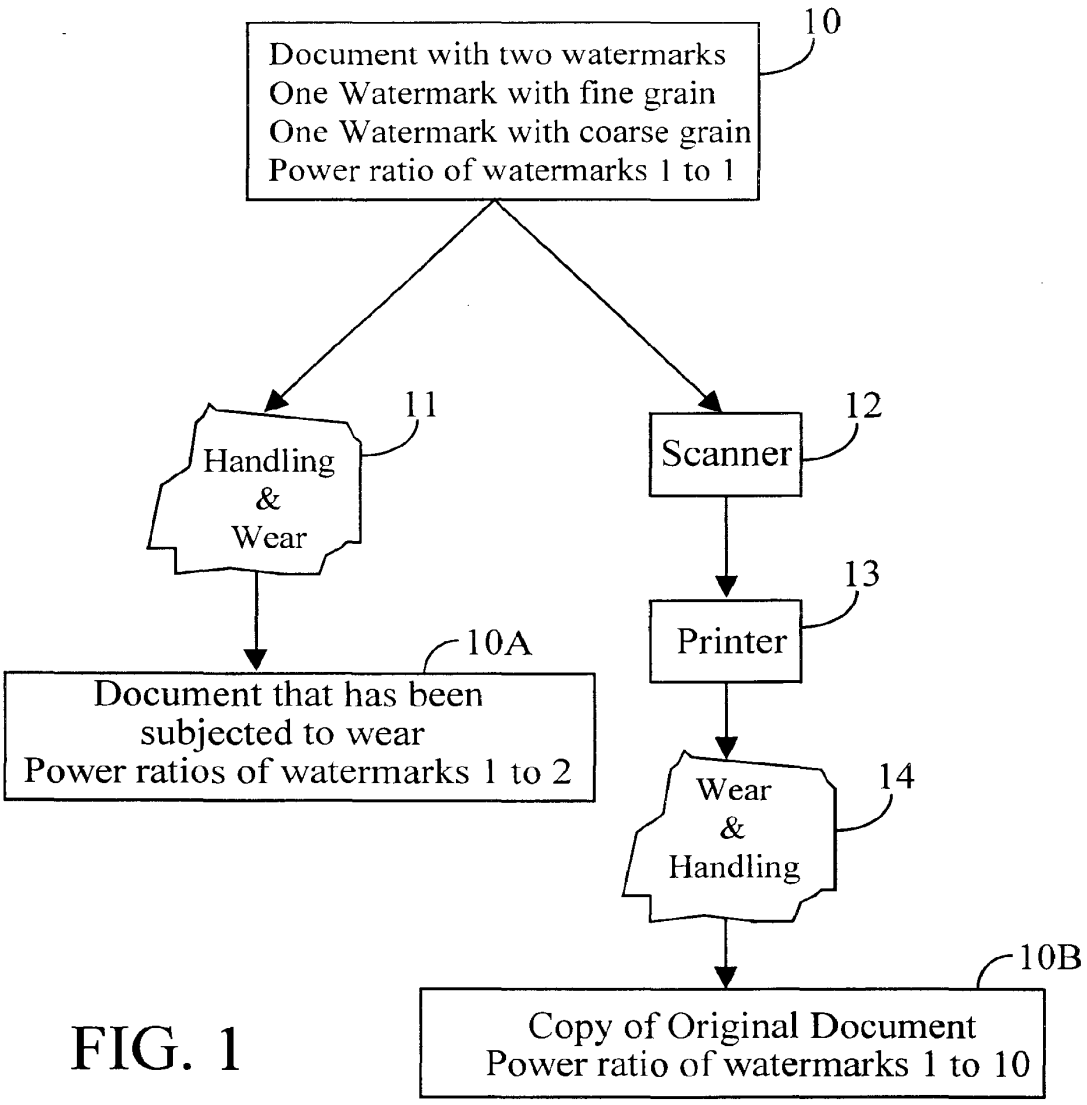

Document with two watermarks
One Watermark with fine grain
One Watermark with coarse grain
Power ratio of watermarks 1 to 1
10
Handling & Wear
11
Scanner
12
Printer
13
Wear & Handling
14
Document that has been subjected to wear
Power ratios of watermarks 1 to 2
10A
Copy of Original Document
Power ratio of watermarks 1 to 10
10B

FIG. 1

FIG. 2A

Watermark with a fine grain (each block of pixels is 3 by 3)

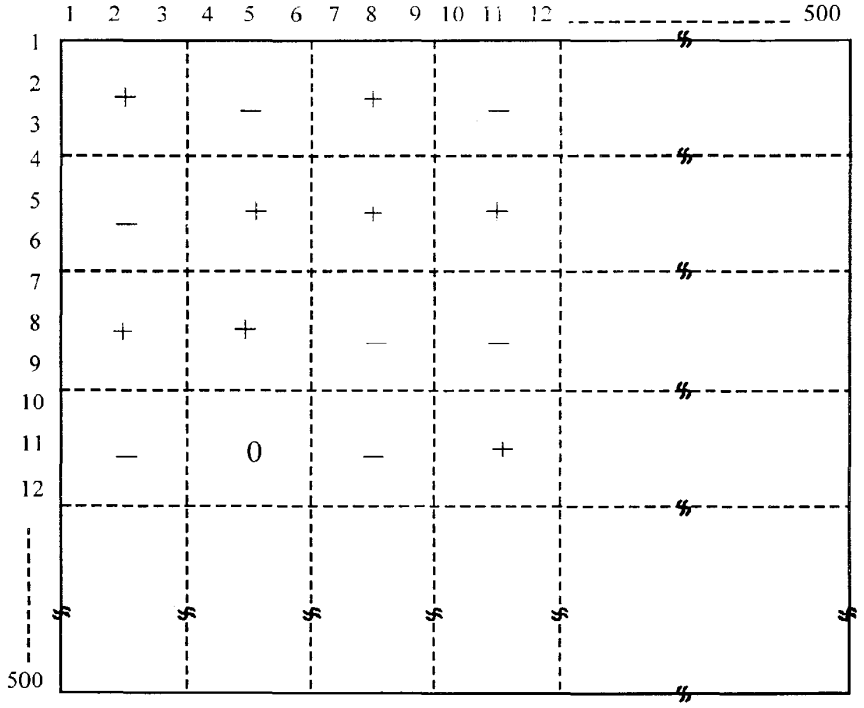


FIG. 2B

Watermark with a coarse grain (each block of pixels is 6 by 6)

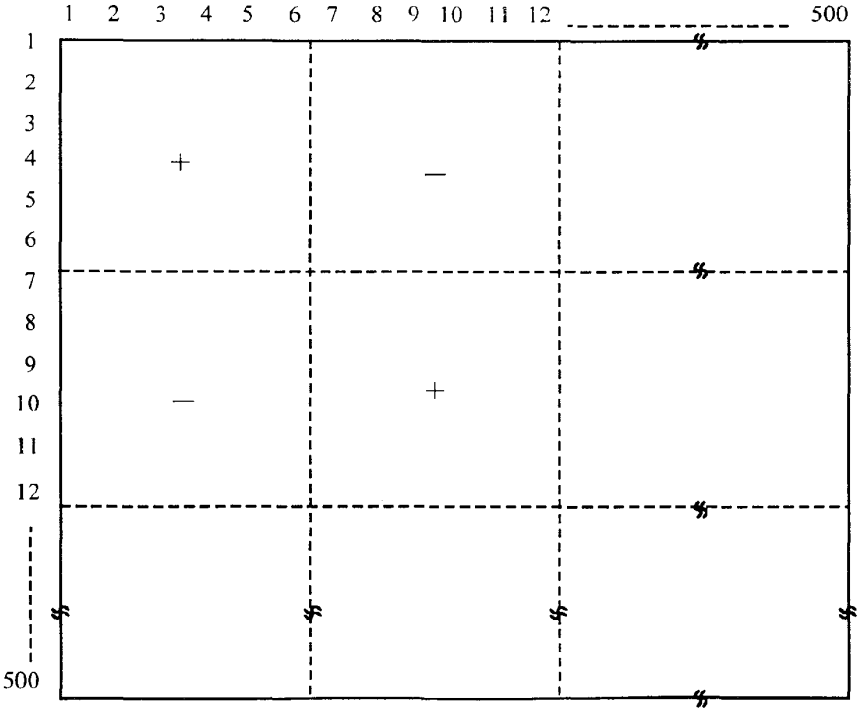


FIG. 3A Geometrically linear assignment of pixels to each bit

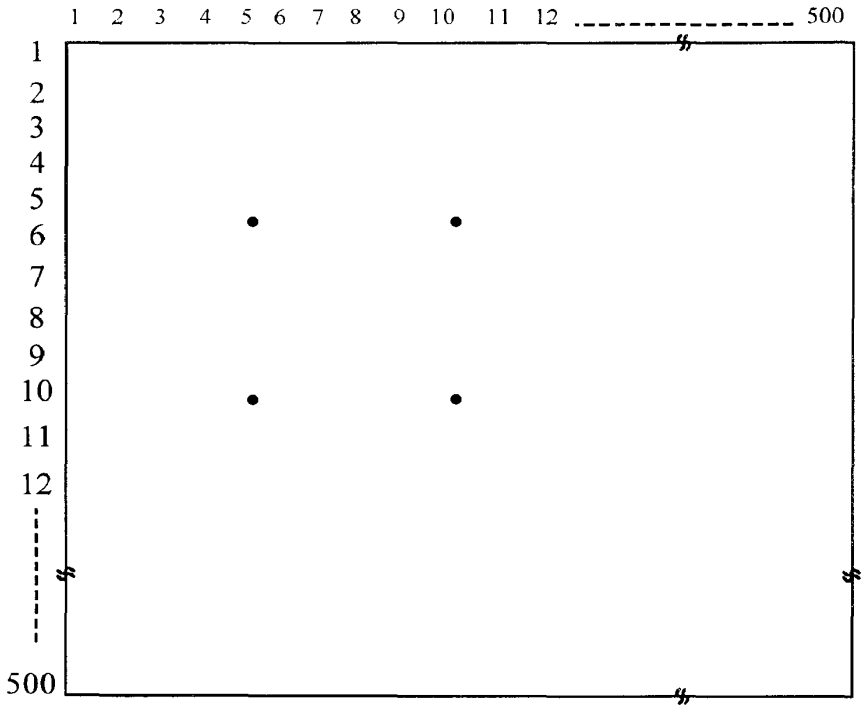
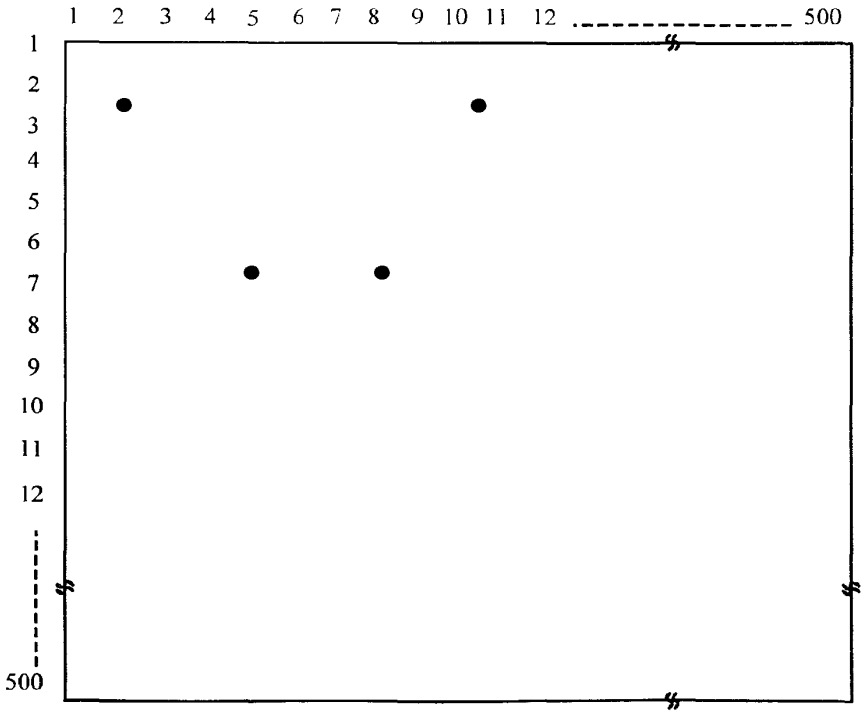


FIG. 3B Geometrically random assignment of pixels to each bit



(1) RGB ⟶ HSI

(2) First Watermark

$$HSI + WM1\Delta \xrightarrow{T} RGB1$$

(3) Second Watermark

$$HSI + \text{Biased } WM2\Delta \longrightarrow RGB2$$

(4) Final image $(RGB1 + RGB2) / 2 = RGBF$

FIG. 4

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
1 March 2001 (01.03.2001)

PCT

(10) International Publication Number
WO 01/15021 A2

(51) International Patent Classification[7]: G06F 17/30

(21) International Application Number: PCT/US00/21452

(22) International Filing Date: 4 August 2000 (04.08.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

09/382,427	24 August 1999 (24.08.1999)	US
09/382,425	24 August 1999 (24.08.1999)	US
09/382,373	24 August 1999 (24.08.1999)	US
09/382,371	24 August 1999 (24.08.1999)	US
09/382,375	24 August 1999 (24.08.1999)	US
09/379,699	24 August 1999 (24.08.1999)	US
09/379,700	24 August 1999 (24.08.1999)	US

(71) Applicant: DIGITALCONVERGENCE.:COM INC. [US/US]; Suite 600, 9101 North Central Expressway, Dallas, TX 75231 (US).

(72) Inventors: PHILYAW, Jeffry, Jovan; 5968 West Northwest Highway, No. 1813, Dallas, TX 75225 (US). MATHEWS, David, Kent; 3438 Livingston Lane, Carrollton, TX 75007 (US).

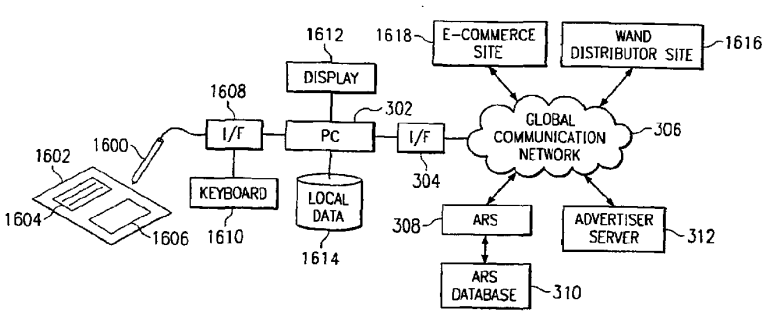
(74) Agents: HOWISON, Gregory, M. et al.; Howison, Chauza, Handley & Arnott, L.L.P., P.O. Box 741715, Dallas, TX 75374-1715 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR ESTABLISHING CONNECTION TO A REMOTE LOCATION ON A COMPUTER NETWORK



(57) Abstract: A method for establishing a connection between a user (302) and a remote location (312) on a computer network (306) is provided, including multiple embodiments. A web browser may be opened and launched in response to an audible signal (111) received at the user location. The web browser automatically retrieves and displays information associated with the received audio signal from a remote site. A bar code (1606) which has no embedded routing information may be scanned to extract bar code information (1802). The bar code information is then utilized to obtain routing information (2102) which directs a user to the remote location. A web browser may be launched by a non-browser input which is correlated to simulate a predetermined browser input. The web browser automatically retrieves and displays information from the remote site. A unique code which has no embedded routing information may be received at the user location. Network routing information is then associated with the unique information and utilized to direct a user to the remote location. An existing machine-readable code on a product may be scanned to extract a product code embedded therein which includes no routing information. The code is then associated with routing information to direct a user to the remote location. A product code having product information is disposed in close association with the subject product. The product information is then extracted from the code and network routing information is associated with the product information. A tool (1600) having a unique ID (1804) is utilized in conjunction with a user computer. In response to utilizing the tool, the user is connected to a remote location associated with the unique ID of the tool.

WO 01/15021 A2

Published:
— *Without international search report and to be republished upon receipt of that report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHOD AND APPARATUS FOR ESTABLISHING CONNECTION TO A REMOTE LOCATION ON A COMPUTER NETWORK

TECHNICAL FIELD OF THE INVENTION

The present invention is related to a method of computer control. In one aspect, it relates to a method for utilizing an existing product code to establish a connection to a remote location on a global computer network.

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a Continuation-in-Part of pending U.S. Patent Application Serial No. 09/378,221 (Atty Dkt. No. PHLY-24,669) entitled "METHOD AND APPARATUS FOR ACCESSING A REMOTE LOCATION BY SCANNING AN OPTICAL CODE," filed on August 19, 1999, which is a Continuation-in-Part of the following two pending U.S. Patent Applications: Serial No. 09/151,471 (Atty Dkt. No. PHLY-24,397) entitled "METHOD FOR INTERFACING SCANNED PRODUCT INFORMATION WITH A SOURCE FOR THE PRODUCT OVER A GLOBAL NETWORK," and Serial No. 09/151,530 (Atty Dkt. No. PHLY-24,398) entitled "METHOD FOR CONTROLLING A COMPUTER WITH AN AUDIO SIGNAL," both filed on September 11, 1998.

BACKGROUND OF THE INVENTION

With the growing numbers of computer users connecting to the "Internet," many companies are seeking the substantial commercial opportunities presented by such a large user base. For example, one technology which exists allows a television ("TV") signal to trigger a computer response in which the consumer will be guided to a personalized web page. The source of the triggering signal may be a TV, video tape recorder, or radio. For example, if a viewer is watching a TV program in which an advertiser offers viewer voting, the advertiser may transmit a unique signal within the television signal which controls a program known as a "browser" on the viewer's computer to automatically display the advertiser's web page. The viewer then simply makes a selection which is then transmitted back to the advertiser.

In order to provide the viewer with the capability of responding to a wide variety of companies using this technology, a database of company information and Uniform Resource Locator ("URL") codes is necessarily maintained in the viewer's computer, requiring continuous updates. URLs are short strings of data that identify resources on the Internet: documents, images, downloadable files, services, electronic mailboxes, and other resources. URLs make resources available under a variety of naming schemes and access methods such as HTTP, FTP, and Internet mail, addressable in the same simple way. URLs reduce the tedium of "login to this server, then issue this magic command ..." down to a single click. The Internet uses URLs to specify the location of files on other servers. A URL includes the type of resource being accessed (e.g., Web, gopher, FTP), the address of the server, and the location of the file. The URL can point to any file on any networked computer. Current technology requires the viewer to perform periodic updates to obtain the most current URL database. This aspect of the current technology is cumbersome since the update process requires downloading information to the viewer's computer. Moreover, the likelihood for error in performing the update, and the necessity of redoing the update in the event of a later computer crash, further complicates the process. Additionally, current technologies are limited in the number of companies which may be stored in the database. This is a significant limitation since world-wide access presented by the Internet and the increasing

number of companies connecting to perform on-line E-commerce necessitates a large database.

SUMMARY OF THE INVENTION

The present invention disclosed and claimed herein, in one aspect thereof, comprises a system and method for launching a web browser on a network comprising a computer having an all new input interface and a communication interface coupled to a computer network; said audio input coupled to the audio output of a source for receiving an audio signal having encoded therein a unique code that is associated with a predetermined destination on the network; and a program operable on said computer responsive to receipt and decoding of the audio signal received from the source for interacting with connections of the computer to a web site available on the computer network wherein the location of the predetermined destination is not stored in a computer.

In another aspect, the present invention comprises a method for interfacing a user location on a network to a destination location on the network. A bar code having product information contained therein relating to an associated product is first scanned with a scanner, which bar code has no network routing information contained therein. The product information contained within the bar code is then extracted. Routing information over the network from the user location to the destination location is then defined in response to the steps of scanning and extracting, which routing information defines the location of the destination location on the network. With the routing information, the user location is then connected to the destination location in accordance with the defined routing information.

In yet another aspect, the present invention comprises a method for launching a web browser application on a user's computer. A browser application is provided on the user's computer that is launchable in response to predetermined browser inputs being received by the user's computer. A non-browser input is provided that is not a portion of the set of predetermined browser inputs. This non-browser is correlated to the input to simulate one or more of the predetermined browser inputs as a simulated browser input. The web browser is then launched on the user's computer with this simulated browser input to access information on a network.

In a further aspect, the present invention comprises a method for interconnecting a user's location to a destination location on a network. The unique information is received at the user's location, which unique information has no associated routing information embedded therein. Network routing information is associated with the received unique information in response to receipt thereof. The user's location is then interconnect to the destination location across the network in accordance with the associated routing associated therewith in the step of associating.

In a still further aspect, the present invention comprises a method for providing an interconnection relationship between a product and a desired location on a global communications network. A machine readable product code is disposed on the product machine readable product code, the machine readable product code having encoded product information contained therein. The product code has no routing information embedded therein which would allow the product code, in and of itself, to cause routing to the desired location over any path on the network. The machine readable code is read and decoded. The extracted product code is then converted for routing information over the network to the desired location, which routing information defines the manner by which a user or a computer at a user location wherein the machine readable code was read can communicate with the desired location via an interconnection therewith.

In another aspect, the present invention comprises a method for utilizing a product code having product information contained therein for interfacing over a network. The product information is extracted from the product code, which product code is disposed on or in close association with an associated product. In response to this extraction, network routing information is associated with the product code information.

In yet another aspect, the present invention comprises a method for accessing information over a network. A tool is utilized in conjunction with an operation on a user's processor at a user location on the network. The tool has associated therewith a unique tool ID. In response to utilizing the tool, the user's location is interconnected on the network to a predetermined destination at a remote location on the network, which destination has an association with the unique ID of the tool.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention and the advantages thereof, reference is now made to the following description taken in conjunction with the accompanying Drawings in which:

FIGURE 1 illustrates a block diagram of the preferred embodiment;

FIGURE 2 illustrates the computer components employed in this embodiment;

FIGURE 3 illustrates system interactions over a global network;

FIGURES 4a-4e illustrate the various message packets transmitted between the source PC and network servers used in the preferred embodiment; and

FIGURE 5 is a flowchart depicting operation of the system according to the preferred embodiment.

FIGURE 6 illustrates a flowchart of actions taken by the Advertiser Reference Server ("ARS") server;

FIGURE 7 illustrates a flowchart of the interactive process between the source computer and ARS;

FIGURE 8 illustrates a web browser page receiving the modified URL/advertiser product data according to the preferred embodiment;

FIGURE 9 illustrates a simplified block diagram of the disclosed embodiment;

FIGURE 10 illustrates a more detailed, simplified block diagram of the embodiment of FIGURE 9;

FIGURE 11 illustrates a diagrammatic view of a method for performing the routing operation;

FIGURE 12 illustrates a block diagram of an alternate embodiment utilizing an optical region in the video image for generating the routing information;

FIGURE 13 illustrates a block diagram illustrating the generation of a profile with the disclosed embodiment;

FIGURE 14 illustrates a flowchart for generating the profile and storing at the ARS;

FIGURE 15 illustrates a flowchart for processing the profile information when information is routed to a user;

FIGURE 16 illustrates a general block diagram of a disclosed embodiment;

FIGURE 16a illustrates a flowchart depicting the operation wherein the input device is registered;

FIGURE 17 illustrates the conversion circuit of the wedge interface;

FIGURE 18 illustrates a sample message packet transmitted from the user PC to the ARS;

FIGURE 19 illustrates a more detailed block diagram of the routing of the message packets between the various nodes;

FIGURE 20 illustrates a block diagram of a browser window, according to a disclosed embodiment;

FIGURE 21 illustrates a diagrammatic view of information contained in the ARS database;

FIGURE 22 illustrates a flowchart of the process of receiving information for the user's perspective;

FIGURE 23 illustrates a flowchart according to the ARS; and

FIGURE 24 illustrates a flowchart of the process performed at the E-commerce node.

DETAILED DESCRIPTION OF THE INVENTION

Referring now to FIGURE 1, there is illustrated a block diagram of a system for controlling a personal computer ("PC") 112 via an audio tone transmitted over a wireless system utilizing a TV. In the embodiment illustrated in FIGURE 1, there is provided a transmission station 101 and a receive station 117 that are connected via a communication link 108. The transmission station 101 is comprised of a television program source 104, which is operable to generate a program in the form of a broadcast signal comprised of video and audio. This is transmitted via conventional techniques along channels in the appropriate frequencies. The program source is input to a mixing device 106, which mixing device is operable to mix in an audio signal. This audio signal is derived from an audio source 100 which comprises a coded audio signal which is then modulated onto a carrier which is combined with the television program source 104. This signal combining can be done at the audio level, or it can even be done at the RF level in the from of a different carrier. However, the preferred method is to merely sum the audio signal from the modulator 102 into the audio channel of the program that is generated by the television program source 104. The output thereof is provided from the mixing device 106 in the form of broadcast signal to an antenna 107 , which transmits the information over the communication link 108 to an antenna 109 on the receive side.

On the receive side of the system, a conventional receiver 110, such as a television is provided. This television provides a speaker output which provides the user with an audible signal. This is typically associated with the program. However, the receiver 110 in the disclosed embodiment, also provides an audio output jack, this being the type RCA jack. This jack is utilized to provide an audio output signal on a line 113 which is represented by an audio signal 111. This line 113 provides all of the audio that is received over the communication link 108 to the PC 112 in the audio input port on the PC 112. However, it should be understood that, although a direct connection is illustrated from the receiver 110 to the PC 112, there actually could be a microphone pickup at the PC 112 which could pick the audio signal up. In the disclosed embodiment the audio signal generated by the advertiser data input device 100 is audible to the human ear and, therefore, can be heard by the user. Therefore, no special filters are needed to provide this audio to the PC 112.

The PC 112 is operable to run programs thereon which typically are stored in a program file area 116. These programs can be any type of programs such as word processing programs, application programs, etc. In the disclosed embodiment, the program that is utilized in the system is what is referred to as a "browser." The PC 112 runs a browser program to facilitate the access of information on the network, for example, a global communication network known as the "Internet" or the World-Wide-Web ("Web"). The browser is a hypertext-linked application used for accessing information. Hypertext is a term used to describe a particular organization of information within a data processing system, and its presentation to a user. It exploits the computer's ability to link together information from a wide variety of sources to provide the user with the ability to explore a particular topic. The traditional style of presentation used in books employs an organization of the information which is imposed upon it by limitations of the medium, namely fixed sized, sequential paper pages. Hypertext systems, however, use a large number of units of text or other types of data such as image information, graphical information, video information, or sound information, which can vary in size. A collection of such units of information is termed a hypertext document, or where the hypertext documents employ information other than text, hypermedia documents. Multimedia communications may use the Hypertext Transfer Protocol ("HTTP"), and files or formatted data may use the Hypertext Markup Language ("HTML"). This formatting language provides for a mingling of text, graphics, sound, video, and hypertext links by "tagging" a text document using HTML. Data encoded using HTML is often referred to as an "HTML document," an "HTML page," or a "home page." These documents and other Internet resources may be accessed across the network by means of a network addressing scheme which uses a locator referred to as a Uniform Resource Locator ("URL"), for example, "http://www.digital.com."

The Internet is one of the most utilized networks for interconnecting distributed computer systems and allows users of these computer systems to exchange data all over the world. Connected to the Internet are many private networks, for example, corporate or commercial networks. Standard protocols, such as the Transport Control Protocol ("TCP") and the Internet Protocol ("IP") provide a convenient method for communicating across these diverse networks. These protocols dictate how data are formatted and communicated. As a characteristic of the Internet, the protocols are layered in an IP stack. At higher levels

of the IP stack, such as the application layer (where HTTP is employed), the user information is more readily visible, while at lower levels, such as the network level (where TCP/IP are used), the data can merely be observed as packets or a stream of rapidly moving digital signals. Superimposed on the Internet is a standard protocol interface for accessing Web resources, such servers, files, Web pages, mail messages, and the like. One way that Web resources can be accessed is by browsers made by *Netscape*® and *Microsoft Internet Explorer*®.

Referring again now to FIGURE 1, the user can load this program with the appropriate keystrokes such that a browser window will be displayed on a display 118. In one embodiment, the user can run the browser program on the PC 112 such that the browser window is displayed on the display 118. While watching a preferred program, the user can also view display 118. When an audio signal is received by the receiver 110 and the encoded information is contained therein that was input thereto by the advertiser, the PC 112 will then perform a number of operations. The first operation, according to the disclosed embodiment, is to extract the audio information within the received audio signal in the form of digital data, and then transmit this digital data to a defined location on the global communication network via a modem connection 114. This connection will be described hereinbelow. This information will be relayed to a proprietary location and the instructions sent back to the PC 112 as to the location of the advertiser associated with the code, and the PC 112 will then effect a communication link to that location such that the user can view on the display 118 information that the advertiser, by the fact of putting the tone onto the broadcast channel, desires the viewer to view. This information can be in the form of interactive programs, data files, etc. In one example, when an advertisement appears on the television, the tone can be generated and then additional data displayed on the display 118. Additionally, a streaming video program could be played on the PC received over the network, which streaming video program is actually longer than the advertising segment on the broadcast. Another example would be a sports game that would broadcast the tone in order to allow a user access to information that is not available over the broadcast network, such as additional statistics associated with the sports program, etc.

By utilizing the system described herein with respect to the disclosed embodiment of FIGURE 1, an advertiser is allowed the ability to control a user's PC 112 through the use of tones embedded within a program audio signal. As will descried hereinbelow, the disclosed embodiment utilizes particular routing information stored in the PC 112 which allows the encoded information in the received audio signal to route this information to a desired location on the network and then allow other routing information to be returned to the PC 112 for control thereof to route the PC 112 to the appropriate location associated with that code.

Referring now to FIGURE 2, there is illustrated a computer 204, similar to computer 112, connected to display information on display 118. The computer 204 comprises an internal audio or "sound" card 206 for receiving the transmitted audio signal through receive antenna 109 and receiver 110. The sound card 206 typically contains analog-to-digital circuitry for converting the analog audio signal into a digital signal. The digital signal may then be more easily manipulated by software programs. The receiver 110 separates the audio signal from the video signal. A special trigger signal located within the transmitted advertiser audio signal triggers proprietary software running on the computer 204 which launches a communication application, in this particular embodiment, the web browser application located on the PC 204. Coded advertiser information contained within the audio signal is then extracted and appended with the address of a proprietary server located on the communication network. The remote server address is in the form of a URL. This appended data, in addition to other control codes, is inserted directly into the web browser application for automatic routing to the communication network.

The web browser running on PC 204, and communicating to the network with a through an internal modem 208, in this embodiment, transmits the advertiser information to the remote server. The remote server cross-references the advertiser product information to the address of the advertiser server located on the network. The address of the advertiser server is routed back through the PC 204 web browser to the advertiser server. The advertiser product information is returned to PC 204 to be presented to the viewer on display 118. In this particular embodiment, the particular advertiser product information

displayed is contained within the advertiser's web page 212. As mentioned above, the audio signal is audible to the human ear. Therefore the audio signal, as emitted from the TV speakers, may be input to the sound card 206 via a microphone. Furthermore, the audio signal need not be a real-time broadcast, but may be on video tapes, CDs, DVD, or other media which may be displayed at a later date. With the imminent implementation of high definition digital television, the audio signal output from the TV may also be digital. Therefore, direct input into a sound card for A/D purposes may not be necessary, but alternative interfacing techniques to accommodate digital-to-digital signal formats would apply.

Referring now to FIGURE 3, there is illustrated a source PC 302, similar to PCs 204 and 112, connected to a global communication network 306 through an interface 304. In this embodiment, the audio signal 111 is received by PC 302 through its sound card 206. The audio signal 111 comprises a trigger signal which triggers proprietary software into launching a web browser application residing on the PC 302. The audio signal 111 also comprises advertiser product information which is extracted and appended with URL information of an Advertiser Reference Server ("ARS") 308. The ARS 308 is a system disposed on the network that is defined as the location to which data in the audio signal 111 is to be routed. As such, data in the audio signal 111 will always be routed to the ARS 308, since a URL is unique on the network system. Connected to the ARS 308 is a database 310 of product codes and associated manufacturer URLs. The database 310 undergoes a continual update process which is transparent to the user. As companies sign-on, e.g., subscribe, to this technology, manufacturer and product information is added to the database 310 without interrupting operation of the source PC 302 with frequent updates. When the advertiser server address URL is obtained from the ARS database 310, it and the request for the particular advertiser product information is automatically routed back through the web browser on PC 302, over to the respective advertiser server for retrieval of the advertiser product information to the PC 302. It should be noted that although the disclosed invention discusses a global communication network, the system is also applicable to LANs, WANs, and peer-to-peer network configurations. It should be noted that the disclosed architecture is not limited to a single source PC 302, but may comprise a plurality of source PCs, e.g.,

PC 300 and PC 303. Moreover, a plurality of ARS 308 systems and advertiser servers 312 may be implemented, e.g., ARS 314, and advertiser server A 316, respectively.

The information transactions, in general, which occur between the networked systems of this embodiment, over the communication network, are the following. The web browser running on source PC 302 transmits a message packet to the ARS 308 over Path "A." The ARS 308 decodes the message packet and performs a cross-reference function with product information extracted from the received message packet to obtain the address of an advertiser server 312. A new message packet is assembled comprising the advertiser server 312 address, and sent back to the source PC 302 over Path "B." A "handoff" operation is performed whereby the source PC 302 browser simply reroutes the information on to the advertiser server 312 over Path "C," with the appropriate source and destination address appended. The advertiser server 312 receives and decodes the message packet. The request-for-advertiser-product-information is extracted and the advertiser 312 retrieves the requested information from its database for transmission back to the source PC 302 over Path "D." The source PC 302 then processes the information, i.e., for display to the viewer. The optional Path "E" is discussed hereinbelow. It should be noted that the disclosed methods are not limited to only browser communication applications, but may accommodate, with sufficient modifications by one skilled in the art, other communication applications used to transmit information over the Internet or communication network.

Referring now to FIGURE 4a, the message packet 400 sent from the source PC 302 to ARS 308 via Path "A" comprises several fields. One field comprises the URL of the ARS 308 which indicates where the message packet is to be sent. Another field comprises the advertiser product code or other information derived from the audio signal 111, and any additional overhead information required for a given transaction. The product code provides a link to the address of the advertiser server 312, located in the database 310. Yet another field comprises the network address of the source PC 302. In general, network transmissions are effected in packets of information, each packet providing a destination address, a source address, and data. These packets vary depending upon the network transmission protocol utilized for communication. Although the protocols utilized in the disclosed embodiment are of a conventional protocol suite commonly known as TCP/IP, it

should be understood that any protocols providing the similar basic functions can be used, with the primary requirement that a browser can forward the routing information to the desired URL in response to keystrokes being input to a PC. However, it should be understood that any protocol can be used, with the primary requirement that a browser can forward the product information to the desired URL in response to keystrokes being input to a PC. Within the context of this disclosure, "message packet" shall refer to and comprise the destination URL, product information, and source address, even though more than a single packet must be transmitted to effect such a transmission.

Upon receipt of the message packet 400 from source PC 302, ARS 308 processes the information in accordance with instructions embedded in the overhead information. The ARS 308 specifically will extract the product code information from the received packet 400 and, once extracted, will then decode this product code information. Once decoded, this information is then compared with data contained within the ARS advertiser database 310 to determine if there is a "hit." If there is no "hit" indicating a match, then information is returned to the browser indicating such. If there is a "hit," a packet 402 is assembled which comprises the address of the source PC 302, and information instructing the source PC 302 as to how to access, directly in a "handoff" operation, another location on the network, that of an advertiser server 312. This type of construction is relatively conventional with browsers such as *Netscape*® and *Microsoft Internet Explorer*® and, rather than displaying information from the ARS 308, the source PC 302 can then access the advertiser server 312. The ARS 308 transmits the packet 402 back to source PC 302 over Path "B." Referring now to FIGURE 4b, the message packet 402 comprises the address of the source PC 302, the URL of the advertiser server 312 embedded within instructional code, and the URL of the ARS 308.

Upon receipt of the message packet 402 by the source PC 302, the message packet 402 is disassembled to obtain pertinent routing information for assembly of a new message packet 404. The web browser running on source PC 302 is now directed to obtain, over Path "C," the product information relevant to the particular advertiser server 312 location information embedded in message packet 404. Referring now to FIGURE 4c, the message

packet 404 for this transaction comprises the URL of the advertiser server 312, the request-for-product-information data, and the address of the source PC 302.

Upon receipt of the message packet 404 from source PC 302, advertiser server 312 disassembles the message packet 404 to obtain the request-for-product-information data. The advertiser server 312 then retrieves the particular product information from its database, and transmits it over Path "D" back to the source PC 302. Referring now to FIGURE 4d, the message packet 406 for this particular transaction comprises the address of the source PC 302, the requested information, and the URL of the advertiser server 312.

Optionally, the ARS 308 may make a direct request for product information over Path "E" to advertiser server 312. In this mode, the ARS 308 sends information to the advertiser server 312 instructing it to contact the source PC 302. This, however, is unconventional and requires more complex software control. The message packet 408 for this transaction is illustrated in FIGURE 4e, which comprises the URL of the advertiser server 312, the request-for-product-information data, and the address of the source PC 302. Since product information is not being returned to the ARS 308, but directly to the source PC 302, the message packet 408 requires the return address to be that of the source PC 302. The product information is then passed directly to PC 302 over Path "D."

Referring now to FIGURE 5, the method for detecting and obtaining product information is as follows. In decision block 500, a proprietary application running resident on a source computer PC 302 (similar to PC 204) monitors the audio input for a special trigger signal. Upon detection of the trigger signal, data following the trigger signal is decoded for further processing, in function block 502. In function block 504, the data is buffered for further manipulation. In decision block 506, a determination is made as to whether the data can be properly authenticated. If not, program flow continues through the "N" signal to function block 520 where the data is discarded. In function block 522, the program then signals for a retransmission of the data. The system then waits for the next trigger signal, in decision block 500. If properly authenticated in decision block 506, program flow continues through the "Y" signal path where the data is then used to launch the web browser application, as indicated in function block 508. In function block 510, the

web browser receives the URL data, which is then automatically routed through the computer modem 208 to the network interface 304 and ultimately to the network 306. In function block 514, the ARS 308 responds by returning the URL of advertiser server 312 to the PC 302. In function block 516, the web browser running on the source PC 302, receives the advertiser URL information from the ARS 308, and transmits the URL for the product file to the advertiser server 312. In block 518, the advertiser server 312 responds by sending the product information to the source PC 302 for processing.

The user may obtain the benefits of this architecture by simply downloading the proprietary software over the network. Other methods for obtaining the software are well-known; for example, by CD, diskette, or pre-loaded hard drives.

Referring now to FIGURE 6, there is illustrated a flowchart of the process the ARS 308 may undergo when receiving the message packet 400 from the source PC 302. In decision block 600, the ARS 308 checks for the receipt of the message packet 400. If a message packet 400 is not received, program flow moves along the "N" path to continue waiting for the message. If the message packet 400 is received, program flow continues along path "Y" for message processing. Upon receipt of the message packet 400, in function block 602, the ARS 308 decodes the message packet 400. The product code is then extracted independently in function block 604 in preparation for matching the product code with the appropriate advertiser server address located in the database 310. In function block 606, the product code is then used with a lookup table to retrieve the advertiser server 312 URL of the respective product information contained in the audio signal data. In function block 608, the ARS 308 then assembles message packet 402 for transmission back to the source PC 302. Function block 610 indicates the process of sending the message packet 402 back to the source PC 302 over Path "B."

Referring now to FIGURE 7, there is illustrated a flowchart of the interactive processes between the source PC 302 and the advertiser server 312. In function block 700, the source PC 302 receives the message packet 402 back from the ARS 308 and begins to decode the packet 402. In function block 702, the URL of the advertiser product information is extracted from the message packet 402 and saved for insertion into the

message packet 404 to the advertiser server 312. The message packet 404 is then assembled and sent by the source PC 302 over Path "C" to the advertiser server 312, n function block 704. While the source PC 302 waits, in function block 706, the advertiser server 312 receives the message packet 404 from the source PC 302, in function block 708, and disassembles it. The product information location is then extracted from the message packet 404 in function block 710. The particular product information is retrieved from the advertiser server 312 database for transmission back to the source PC 302. In function block 712, the product information is assembled into message packet 406 and then transmitted back to the source PC 302 over Path "D." Returning to the source PC 302 in function block 714, the advertiser product information contained in the message packet 406 received from the advertiser server 312, is then extracted and processed in function block 716.

Referring now to FIGURE 8, after receipt of a trigger signal, a web browser application on a source PC 302 is automatically launched and computer display 800 presents a browser page 802. Proprietary software running on the source PC 302 processes the audio signal data after being digitized through the sound card 206. The software appropriately prepares the data for insertion directly into the web browser by extracting the product information code and appending keystroke data to this information. First, a URL page 804 is opened in response to a Ctrl-O command added by the proprietary software as the first character string. Opening URL page 804 automatically positions the cursor in a field 806 where additional keystroke data following the Ctrl-O command will be inserted. After URL page 804 is opened, the hypertext protocol preamble http:// is inserted into the field 806. Next, URL information associated with the location of the ARS 308 is inserted into field 806. Following the ARS 308 URL data are the characters /? to allow entry of variables immediately following the /? characters. In this embodiment, the variable following is the product information code received in the audio signal. The product code information also provides the cross-reference information for obtaining the advertiser URL from the ARS database 310. Next, a carriage return is added to send the URL/product data and close the window 804. After the message packet 400 is transmitted to the ARS 308 from the source PC 302, transactions from the ARS308, to the source PC 302, to the advertiser server 312, and back to the source PC 302, occur quickly and are transparent to the viewer. At this

point, the next information the viewer sees is the product information which was received from the advertiser server 312.

Referring now to FIGURE 9, there is illustrated a block diagram of a more simplified embodiment. In this embodiment, a video source 902 is provided which is operable to provide an audio output on an audio cable 901 which provides routing information referred to by reference numeral 904. The routing information 904 is basically information contained within the audio signal. This is an encoded or embedded signal. The important aspect of the routing information 904 is that it is automatically output in realtime as a function of the broadcast of the video program received over the video source 902. Therefore, whenever the program is being broadcast in realtime to the user 908, the routing information 904 will be output whenever the producer of the video desires it to be produced. It should be understood that the box 902 representing the video source could be any type of media that will result in the routing information being output. This could be a cassette player, a DVD player, an audio cassette, a CD ROM or any such media. It is only important that this is a program that the producer develops which the user 908 watches in a continuous or a streaming manner. Embedded within that program, at a desired point selected by the producer, the routing information 904 is output.

The audio information is then routed to a PC 906, which is similar to the PC 112 in FIGURE 1. A user 908 is interfaced with the PC to receive information thereof, the PC 906 having associated therewith a display (not shown). The PC 906 is interfaced with a network 910, similar to the network 306 in FIGURE 3. This network 910 has multiple nodes thereon, one of which is the PC 906, and another of which is represented by a network node 912 which represents remote information. The object of the present embodiment is to access remote information for display to the user 908 by the act of transmitting from the video program in block 902 the routing information 904. This routing information 904 is utilized to allow the PC 906 which has a network "browser" running thereon to "fetch" the remote information at the node 912 over the network 910 for display to the user 908. This routing information 904 is in the form of an embedded code within the audio signal, as was described hereinabove.

Referring now to FIGURE 10, there is illustrated a more detailed block diagram of the embodiment of FIGURE 9. In this embodiment, the PC 906 is split up into a couple of nodes, a first PC 1002 and a second PC 1004. The PC 1002 resides at the node associated with the user 908, and the PC 1004 resides at another node. The PC 1004 represents the ARS 308 of FIGURE 3. The PC 1004 has a database 1006 associated therewith, which is basically the advertiser database 310. Therefore, there are three nodes on the network 910 necessary to implement the disclosed embodiment, the PC 1002, the PC 1004 and the remote information node 912. The routing information 904 is utilized by the PC 1002 for routing to the PC 1004 to determine the location of the remote information node 912 on the network 910. This is returned to the PC 1002 and a connection made directly with the remote information node 912 and the information retrieved therefrom to the user 908. The routing information 904 basically constitutes primary routing information.

Referring now to FIGURE 11, there is illustrated a diagrammatic view of how the network packet is formed for sending the primary routing information to the PC 1004. In general, the primary routing information occupies a single field which primary routing information is then assembled into a data packet with the secondary routing information for transfer to the network 910. This is described hereinabove in detail.

Referring now to FIGURE 12, there is illustrated an alternate embodiment to that of FIGURE 9. In this embodiment, the video source 902 has associated therewith an optical region 1202, which optical region 1202 has disposed therein an embedded video code. This embedded video code could be relatively complex or as simple as a grid of dark and white regions, each region in the grid able to have a dark color for a logic "1" or a white region for a logic "0." This will allow a digital value to be disposed within the optical region 1202. A sensor 1204 can then be provided for sensing this video code. In the example above, this would merely require an array of optical detectors, one for each region in the grid to determine whether this is a logic "1" or a logic "0" state. One of the sensed video is then output to the PC 906 for processing thereof to determine the information contained therein, which information contained therein constitutes the primary routing information 904. Thereafter, it is processed as described hereinabove with reference to FIGURE 9.

Referring now to FIGURE 13, there is illustrated a block diagram for an embodiment wherein a user's profile can be forwarded to the original subscriber or manufacturer. The PC 906 has associated therewith a profile database 1302, which profile database 1302 is operable to store a profile of the user 908. This profile is created when the program, after initial installation, requests profile information to be input in order to activate the program. In addition to the profile, there is also a unique ID that is provided to the user 908 in association with the browser program that runs on the PC 906. This is stored in a storage location represented by a block 1304. This ID 1304 is accessible by a remote location as a "cookie" which is information that is stored in the PC 906 in an accessible location, which accessible location is actually accessible by the remote program running on a remote node.

The ARS 308, which basically constitutes the PC 1004 of FIGURE 10, is operable to have associated therewith a profile database 1308, which profile database 1308 is operable to store profiles for all of the users. The profile database 1308 is a combination of the stored in profile database 1302 for all of the PCs 906 that are attachable to the system. This is to be distinguished from information stored in the database 310, the advertiser's database, which contains intermediate destination tables. When the routing information in the primary routing information 904 is forwarded to the ARS 308 and extracted from the original data packet, the lookup procedure described hereinabove can then be performed to determine where this information is to be routed. The profile database 1302 is then utilized for each transaction, wherein each transaction in the form of the routing information received from the primary routing information 904 is compared to the destination tables 310 to determine what manufacturer it is associated with.

The associated ID 1304 that is transmitted along with the routing information in primary routing information 904 is then compared with the profile database 1308 to determine if a profile associated therewith is available. This information is stored in a transaction database 1310 such that, at a later time, for each routing code received in the form of the information in primary routing information 904, there will associated therewith the IDs 1304 of each of the PCs 906. The associated profiles in database 1308, which are stored in association with IDs 1304, can then be assembled and transmitted to a subscriber

as referenced by a subscriber node 1312 on the network 910. The ARS 308 can do this in two modes, a realtime mode or a non-realtime mode. In a realtime mode, each time a PC 906 accesses the advertiser database 310, that user's profile information is uploaded to the subscriber node 1312. At the same time, billing information is generated for that subscriber 1312 which is stored in a billing database 1316. Therefore, the ARS 308 has the ability to inform the subscriber 1312 of each transaction, bill for those transactions, and also provide to the subscriber 1312 profile information regarding who is accessing the particular product advertisement having associated therewith the routing information field 904 for a particular routing code as described hereinabove. This information, once assembled, can then be transmitted to the subscriber 1312 and also be reflected in billing information and stored in the billing information database 1316.

Referring now to FIGURE 14, there is illustrated a flowchart depicting the operation for storing the profile for the user. The program is initiated in a block 1402 and then proceeds to a function block 1404, wherein the system will prompt for the profile upon initiation of the system. This initiation is a function that is set to activate whenever the user initially loads the software that he or she is provided. The purpose for this is to create, in addition to the setup information, a user profile. Once the user is prompted for this, then the program will flow to a decision block 1406 to determine whether the user provides basic or detailed information. This is selectable by the user. If selecting basic, the program will flow to a function block 1408 wherein the user will enter basic information such as name and serial number and possibly an address. However, to provide some incentive to the user to enter more information, the original prompt in function block 1404 would have offers for such things as coupons, discounts, etc, if the user will enter additional information. If the user selects this option, the program from the decision block 1406 to a function block 1410. In the function block 1410, the user is prompted to enter specific information such as job, income level, general family history, demographic information and more. There can be any amount of information collected in this particular function block.

Once all of the information is collected, in either the basic mode or the more specific mode, the program will then flow to a function block 1412 where this information is stored locally. The program then flows to a decision block 1414 to then go on-line to the host or

the ARS 308. In general, the user is prompted to determine whether he or she wants to send this information to the host at the present time or to send it later. If he or she selects the "later" option, the program will flow to a function block 1415 to prompt the user at a later time to send the information. In the disclosed embodiment, the user will not be able to utilize the software until the profile information is sent to the host. Therefore, the user may have to activate this at a later time in order to connect with the host.

If the user has selected the option to upload the profile information to the host, the program will flow to the function block 1416 to initiate the connect process and then to a decision block 1418 to determine if the connection has been made. If not, the program will flow along a "N" path to a time to decision block 1420 which will time to an error block 1422 or back to the input of the connect decision block 1418. The program, once connected, will then flow along a "Y" path from decision block 1418 to a function block 1428 to send the profile information with the ID of the computer or user to the host. The ID is basically, as described hereinabove, a "cookie" in the computer which is accessed by the program when transmitting to the host. The program will then flow to a function block 1430 to activate the program such that it, at later time, can operate without requiring all of the set up information. In general, all of the operation of this flowchart is performed with a "wizard" which steps the user through the setup process. Once complete, the program will flow to a Done block 1432.

Referring now to FIGURE 15, there is illustrated a flowchart depicting the operation of the host when receiving a transaction. The program is initiated at a start block 1502 and then proceeds to decision block 1504, wherein it is determined whether the system has received a routing request, i.e., the routing information 904 in the form of a tone, etc., embedded in the audio signal as described hereinabove with respect to FIGURE 9. The program will loop back around to the input of decision block 1504 until the routing request has been received. At this time, the program will flow along the "Y" path to a function block 1506 to receive the primary routing information and the user ID. Essentially, this primary routing information is extracted from the audio tone, in addition to the user ID. The program then flows to a function block 1508 to look up the manufacturer URL that corresponds to the received primary routing information and then return the necessary

command information to the originating PC 108 in order to allow that PC to connect to the destination associated with the primary routing information. Thereafter, the program will flow to a function block 1510 to update the transaction database 1310 for the current transaction. In general, the routing information 904 will be stored as a single field with the associated IDs. The profile database, as described hereinabove, has associated therewith detailed profiles of each user on the system that has activated their software in association with their ID. Since the ID was sent in association with the routing information, what is stored in the transaction database is the routing code, in association with all of the IDs transmitted to the system in association with that particular routing code. Once this transaction database has been updated, as described hereinabove, the transactions can be transferred back to the subscriber at node 312 with the detailed profile information from the profile database 1308.

The profile information can be transmitted back to the subscriber or manufacturer in the node 312 in realtime or non-realtime. A decision block 1512 is provided for this, which determines if the delivery is realtime. If realtime, the program will flow along a "Y" path to a function block 1514 wherein the information will be immediately forwarded to the manufacturer or subscriber. The program will then flow to a function block 1516 wherein the billing for that particular manufacturer or subscriber will be updated in the billing database 1316. The program will then flow into an End block 1518. If it was non-realtime, the program moves along the "N" path to a function block 1520 wherein it is set for a later delivery and it is accrued in the transaction database. In any event, the transaction database will accrue all information associated with a particular routing code.

With a realtime transaction, it is possible for a manufacturer to place an ad in a magazine or to place a product on a shelf at a particular time. The manufacturer can thereafter monitor the times when either the ads are or the products are purchased. Of course, they must be scanned into a computer which will provide some delay. However, the manufacturer can gain a very current view of how a product is moving. For example, if a cola manufacturer were to provide a promotional advertisement on, for example, television, indicating that a new cola was going to be placed on the shelf and that the first 1000 purchasers, for example, scanning their code into the network would receive some benefit,

such as a chance to win a trip to some famous resort in Florida or some other incentive, the manufacturer would have a very good idea as to how well the advertisement was received. Further, the advertiser would know where the receptive markets were. If this advertiser, for example, had placed the television advertisement in ten cities and received overwhelming response from one city, but very poor response from another city, he would then have some inclination to believe that either one poor response city was not a good market or that the advertising medium he had chosen was very poor. Since the advertiser can obtain a relatively instant response and also content with that response as to the demographics of the responder, very important information can be obtained in a relatively short time.

It should be noted that the disclosed embodiment is not limited to a single source PC 302, but may encompass a large number of source computers connected over a global communication network. Additionally, the embodiment is not limited to a single ARS 308 or a single advertiser server 312, but may include a plurality of ARS and advertiser systems, indicated by the addition of ARS 314 and advertiser server A 316, respectively. It should also be noted that this embodiment is not limited only to global communication networks, but also may be used with LAN, WAN, and peer-to-peer configurations.

It should also be noted that the disclosed embodiment is not limited to a personal computer, but is also applicable to, for example, a Network Computer ("NetPC"), a scaled-down version of the PC, or any system which accommodates user interaction and interfaces to information resources.

One typical application of the above noted technique is for providing a triggering event during a program, such as a sport event. In a first example, this may be generated by an advertiser. One could imagine that, due to the cost of advertisements in a high profile sports program, there is a desire to utilize this time widely. If, for example, an advertiser contracted for 15 seconds worth of advertising time, they could insert within their program a tone containing the routing information. This routing information can then be output to the user's PC which will cause the user's PC to, via the network, obtain information from a remote location typically controlled by the advertiser. This could be in the form of an advertisement of a length longer than that contracted for. Further, this could be an

interactive type of advertisement. An important aspect to the type of interaction between the actual broadcast program with the embedded routing information and the manufacturer's site is the fact that there is provided in the information as to the user's PC and a profile of the user themselves. Therefore, an advertiser can actually gain realtime information as to the number of individuals that are watching their particular advertisement and also information as to the background of those individuals, profile information, etc. This can be a very valuable asset to an advertiser.

In another example, the producer of the program, whether it be an on-air program, a program embedded in a video tape, CD-ROM, DVD, or a cassette, can allow the user to automatically access additional information that is not displayed on the screen. For example, in a sporting event, various statistics can be provided to the user from a remote location, merely by the viewer watching the program. When these statistics are provided, the advertiser can be provided with profile information and background information regarding the user. This can be important when, for example, the user may record a sports program. If the manufacturer sees that this program routing code is being output from some device at a later time than the actual broadcast itself, this allows the advertisers to actually see that their program is still being used and also what type of individual is using it. Alternatively, the broadcaster could determine the same and actually bill the advertiser an additional sum for a later broadcast. This is all due to the fact that the routing information automatically, through a PC and a network, will provide an indication to the advertiser for other intermediary regarding the time at which the actual information was broadcast.

The different type of medium that can be utilized with the above embodiment are such things as advertisements, which are discussed hereinabove, contests, games, news programs, education, coupon promotional programs, demonstration media (demos), photographs, all of which can be broadcast on a private site or a public site. This all will provide the ability to allow realtime interface with the network and the remote location for obtaining the routed information and also allow for realtime billing and accounting.

Referring now to FIGURE 16, there is illustrated a general block diagram of a disclosed embodiment. A bar code scanning wand 1600 is provided by a wand distributor

to customers and is associated with that distributor via a wand ID stored therein. The wand 1600 is either sold or freely distributed to customers for use with their personal computing systems. Since more and more products are being sold using bar codes (representing UPCs, ISBNs and EANs), it can be appreciated that a user having the wand 1600 can scan bar codes of a multitude of products in order to obtain more information. Information about these products can be made immediately available to the user from the manufacturer for presentation by the user's computer 302. Beyond simply displaying information about the product in which the user is interested, the wand distributor may include additional advertising information for display to the user such as information about other promotions or products provided or sold by the wand distributor. Similarly, advertisers may provide catalogs of advertisements or information in newspapers or periodicals where the user simply scans the bar code associated with the advertisement using the wand 1600 to obtain further information. There is provided a paper source 1602 having contained thereon an advertisement 1604 and an associated bar code 1606. (Note that the disclosed concept is not limited to scanning of bar codes 1606 from paper sources 1602, but is also operable to scan a bar code 1606 on the product itself. Also, the wand 1600 can be any type of device that will scan any type of image having information encoded therein.) Further description of applications and operation of the bar code scanning wand 1600 and the use of the bar code information is found in pending U.S. Patent Application, Serial No. 09/151,471 filed on September 11, 1998 and entitled "METHOD FOR INTERFACING SCANNED PRODUCT INFORMATION WITH A SOURCE FOR THE PRODUCT OVER A GLOBAL NETWORK," which application is incorporated by reference in its entirety herein.

After obtaining the wand 1600 from the wand distributor, the user connects the wand 1600 to their PC 302. During a scanning operation, wand 1600 reads bar code data 1606 and the wand ID into a "wedge" interface 1608 for conversion into keyboard data, which keyboard data is passed therefrom into the keyboard input port of PC 302. The importance of the wand ID will be discussed in more detail hereinbelow.

The wedge interface 1608 is simply an interface box containing circuitry that accommodates inputs from both the scanning wand 1600 and a computer keyboard 1610. This merely allows the information scanned by the wand 1600 to be input into the PC302.

In the disclosed embodiment, the wedge interface 1608 will convert any information. The data output from the wand 1600 is passed into the wedge interface 1608 for conversion into keyboard data which is readily recognizable by the PC 302. Therefore, the wand 1600 is not required to be connected to a separate port on the PC 302. This data is recognized as a sequence of keystrokes. However, the output of the wand 1600 can be input in any manner compatible with the PC 302. When not receiving scanner data, the wedge interface 1608 simply acts as a pass-through device for keyboard data from the keyboard 1610. In any case, the information is ultimately processed by a processor in the PC 302 and can be presented to the user on a display 1612. The wedge interface is operable to provide a decoding function for the bar code and conversion thereof to keystroke input data.

In operation, the product code of a product is provided in the form of a bar code 1606. This bar code 1606 is the "link" to a product. The disclosed embodiment is operable to connect that product information contained in the bar code 1606 with a web page of the manufacturer of that product by utilizing the bar code 1606 as the product "identifier." The program operating on the PC 302 provides routing information to the ARS 308 after launching the browser on the PC 302 and connecting to the ARS 308 over the GCN 306, which ARS 308 then performs the necessary steps to cause the browser to connect to the manufacturer web site, while also providing for an accounting step, as will be described in more detail hereinbelow.

The bar code 1606 by itself is incompatible with any kind of network for the purposes of communication therewith. It is primarily provided for a retail-type setting. Therefore, the information contained in the bar code 1606, by itself, does not allow for anything other than identification of a product, assuming that one has a database 1614 containing information as to a correlation between the product and the bar code 1606.

The wedge interface 1608 is operable to decode the bar code 1606 to extract the encoded information therein, and append to that decoded bar code information relating to an ID for the wand 1600. This information is then forwarded to the ARS by the resident program in the PC 302. This is facilitated by intermediate routing information stored in the program indicating to which node on the GCN 306 the scanned bar code information is to

be sent, i.e., to the ARS 308. It is important to note that the information in the bar code 1606 must be converted from its optical image to numerical values which are then ultimately input to the keyboard input port of PC 302 and converted into data compatible with communication software residing on the PC 302 (in this case, HTML language for insertion into a browser program). When the scanned information is input to the PC 302, the resident program launches the browser program and then assembles a communication packet comprised of the URL of the ARS 308, the wand ID and the user ID. If another type of communications program were utilized, then it would have to be converted into language compatible with that program. Of course, a user could actually key in the information on the bar code 102 and then append the appropriate intermediate routing information thereafter. As will be described hereinbelow, the intermediate routing information appended thereto is the URL of the ARS 308 disposed on the GCN 306.

As part of the configuration for using the wand 1600, the PC 302 hosts wand software which is operable to interpret data transmitted from the wand 1600, and to create a message packet having the scanned product information and wand ID, routing information, and a user ID which identifies the user location of the wand 1600. The wand software loads at boot-up of the PC 302 and runs in the background. In response to receiving a scanned bar code 1606, the wedge interface outputs a keystroke code (e.g., ALT-F10) to bring the wand program into the foreground for interaction by the operating system. The wand program then inserts the necessary information into the browser program. The message packet is then transmitted to interface 304 across the global communication network 306 to the ARS 308. The ARS 308 interrogates the message packet and performs a lookup function using the ARS database 310. If a match is found between particular parameters of the message packet, a return message packet is sent back to the PC 302 for processing.

The wand program running on PC 302 functions to partition the browser window displayed to the user into several individual areas. This is for the purpose of preparing to present to the user selected information in each of the individual areas (also called "framing"). The selected information comprises the product information which the user requested by scanning the bar code 1606 using the wand 1600, information about the wand distributor which establishes the identity of the company associated with that particular

wand 1600, and at least one or more other frames which may be advertisements related to other products that the wand distributor sells. Note that the advertisements displayed by the wand distributor may be related to the product of interest or totally unrelated. For example, if a user scans the bar code 1606 of a Company A soda, the wand distributor may generate an advertisement of a new soft drink being marketed by Company A, that it sells. On the other hand, the wand distributor may also structure the display of information to the user such that a user requesting product information of a Product X may get the requested information of Product X along with advertisements for a competing item Product Y. Essentially, the wand distributor is free to generate any advertisement to the user in response to the user requesting product information.

The return message packet transmitted from the ARS 308 to the PC 302 is then transmitted back across the GCN 306 to the advertiser server 312. The advertiser server 312 restructures the message packet and appends the particular product information for transmission back to the PC 302. Upon receiving the particular advertiser information from advertiser server 312, the PC 302 then retransmits a message to the wand distributor site 1616 and E-commerce site 1618 to obtain the information that needs to be framed in the browser window displayed to the user.

Therefore, the wand 1600 is associated with the wand distributor by way of a wand ID such that scanning a product bar code 1606 in order to obtain information about that particular product generates one or more responses from one or more remote sites disposed on the GCN 306. Stored in the wand 1600 is the wand ID which establishes its relationship to the wand distributor. Proprietary wand software running on the PC 302 operates to decode scanned bar code information and the wand ID received from the wand 1600 and wedge interface 1608, and also provides a unique user ID for establishing the location of the user of the wand 1600. The wand software also assembles message packets and works in conjunction with the on-board communication software (e.g., a browser) to automatically route the message packets across the GCN 306 such that the one or more remote sites disposed on the GCN 306 return information to be framed for presentation to the user.

Referring now to FIGURE 16a, there is illustrated an addition to the flowchart of FIGURE 14 for registering an input device, for one embodiment of the disclosure, with the ARS 308 in conjunction with a user ID. As described hereinabove, each user, when they generated their user profile when first contacting the ARS 308 to activate their software, defines a user ID for storage in their PC resident software. This user ID, as also described hereinabove, is utilized to identify the PC 302 to the ARS 308 whenever making a transaction. This is utilized for, among other things, updating profile information in the database 310. However, when an input device, such as the input device in the form of a wand 1600, is utilized, there are a number of ways to identify this wand 1600. One can be with the use of a unique wand ID associated with the wand 1600, as will be described hereinbelow. The other is to create a wand ID in the software program operating in the PC 302 running the PC resident background program as described hereinabove. In a situation where the ID is created in the program, reference is made to block 1410 of FIGURE 14 and subsequent thereto. After the user has entered very specific information, jobs, history and demographics, etc., the system will then prompt the individual for information regarding the input device, as indicated by a function block 1630. This prompt will basically inquire as to whether the user has an input device for reading machine readable code or for sensing audible signals. It may desire to know whether the user has received an audible code from a broadcast system such as a TV, or the wand 1600 is being utilized to scan machine readable code in the form of a bar code. The program block will then flow to a function block 1632 wherein various input device types are presented to the user on their display returned over the network 306 from the ARS 308, and then the user selects which type of input device is attached to their computer. For example, it might be a scanning wand, it might be an audio input device, or any other type of device that allows for input of information. For example, it is conceivable that the unique wand ID is that of some type of tool being utilized in conjunction with the program as an input device. The unique wand ID could even be the idea of a disk drive wherein the wand ID would be generalized as a tool ID.

Once the type of input device is input, then the prompt will query what type of device is attached, such as the type of scanning wand 1600. The system will then ask for the distributor of the wand 1600. Of course, as described hereinabove, the wand 1600 could have embedded therein a unique wand ID which would remove the need for inputting the

distributor, as the ARS 308 will recognize this particular wand ID. However, in the situation wherein the wand ID is not disposed within the wand 1600, then the user will have to view the wand or tool 1600 to determine the distributor thereof. This is what input information would then be noted in the user profile in the ARS 308. This allows the ARS 308, at a later time, to match up the user ID, requiring no other user ID, with their user profile to determine what distributor distributed their wand 1600, such that information can be returned associated with that distributor, as will be described hereinbelow. Therefore, the combination of a user ID and a corresponding entry in the database of the ARS 308 will constitute the wand ID or tool ID.

Referring now to FIGURE 17, there is illustrated a conversion circuit of the wedge interface. A microcontroller 1700 provides conversion of the data from the wand 1600 and controls interfacing of the keyboard 1610 and wand 1600 with the PC 302. The microcontroller 1700 has contained therein a memory 1702 for storing a wand ID (or even a URL for some applications) or it can have external memory. There are provided a plurality of wand interfaces 1704 to the wand 1600, a plurality of PC interfaces 1706 to the PC 302, and plurality of keyboard interfaces 1708 to the keyboard 106. In general, the wand interfaces 1704 comprise a serial data line, a ground line, and a power line. Similarly, the keyboard interfaces 1708 comprise a serial data line, a ground line, a clock line, and a power line. The PC 302 provides a clock line, a power line, a serial data, and a ground line for input to the microcontroller 1700. The microcontroller 1700 is operable to receive signals from the keyboard 1610 and transfer the signals to the PC 302 as keyboard signals. Operation with the keyboard 1610 is essentially a "pass-through" procedure. Data output from the keyboard 1610 is already in keyboard format, and therefore requires no conversion by the wedge interface 1608. With respect to the wand 1600, the serial data is not compatible with a keyboard 1610 and, therefore, it must be converted into a keyboard format in order to allow input thereof to the keyboard input of the PC 302.

The microcontroller 1700 performs the conversion function after decoding this bar code information, and conversion of this bar code information into an appropriate stream of data which is comprised of the bar code information and the appended wand ID (or even a URL in some applications). This appended wand ID (and/or URL) will be pre-stored in the

memory 1702 and is programmable at the time of manufacture. It is noted that the memory 1702 is illustrated as being contained within the microcontroller 1702 to provide a single chip solution. However, this could be external memory that is accessible by the microcontroller 1702. Therefore, the microcontroller 1700 provides an interface between the wand 1600 and the keyboard 1610 to the PC 302 which allows the wand 1600 to receive coded information and convert it to keyboard strokes or, alternatively, to merely pass-through the keystrokes from the keyboard 1610. Therefore, the user need not install any type of plug-in circuit board into the motherboard of the PC 302 in order to provide an interface to the wand 1600; rather, the user need only utilize the already available keyboard port in order to input the appropriate data into the system.

In this particular disclosed embodiment, the microcontroller 1700 comprises a PIC16C73 microcontroller by *Microchip Technologies*™. The PIC16C73 device is a low cost CMOS 8-bit microcontroller with an integrated analog-to-digital converter. The PIC16C73 device, as illustrated in the disclosed embodiment, has 192 bytes of RAM and 4k x 4 of EPROM memory. The microcontroller 1700 can accommodate asynchronous or synchronous inputs from input devices connected to it. In this disclosed embodiment, communication to the keyboard 1610 is synchronous while it is asynchronous when communicating with wand 1600.

It should be noted that, although in this particular embodiment bar code information of the bar code 1606 is input into the keyboard input port of the PC 302, disclosed methods may also be advantageously utilized with high speed port architectures such as Universal Serial Bus ("USB") and IEEE 1394.

Bar codes are structured to be read in either direction. Timing considerations need to be addressed because of the variety of individuals scanning the bar code introduce a wide variety of scan rates. Bar codes use bars of varying widths. The presence of a black bar generates a positive pulse, and the absence of a black bar generates no pulse. Each character of a conventional bar code has associated therewith seven pulses or bars. Depending on the width of the bars, the time between pulses varies. In this disclosed embodiment, the interface circuitry 1608 performs a "running" calculation of the scan time based upon the

rising edge of the pulses commencing with the leader or header information. The minimum and maximum scans times are calculated continuously in software with the interface 1608 during the scanning process to ensure a successful scan by the user.

Referring now to FIGURE 18, there is illustrated a sample message packet transmitted from the user's PC 302 to the ARS. The message packet 1800 comprises a number of bits of information including the bar code information 1802 obtained from the user scanning the bar code with the wand 1600; the wand ID 1804 which is embedded in a memory in the wand 1600 and identifies it with a particular wand distributor; and a user ID 1806 which is derived from the software running on the PC 302 and which identifies uniquely with the user location. Note that the message packet includes other necessary information for the proper transmission for point to point.

Referring now to FIGURE 19, there is illustrated a more detailed block diagram of the routing of the message packets in order to present the framed information to the user. As is mentioned hereinabove, when the user scans a bar code 1606 using the wand 1600, a wand program running on the user PC 302 is operable to interpret the information output by the wand 1600 and generate a message packet for transmission over the GCN 306. The wand program assembles the message packet such that it is directed to the ARS 308 disposed on the GCN 306. The message packet contains several pieces of information including the wand ID 1804 which links it to the wand distributor, the user ID 1806 which identifies the particular user using the wand 1600, and bar code information describing a particular product of interest to the user. This message from the PC 302 is transmitted over a path 1900 to the ARS 308 where the ARS database 310 is accessed to cross reference the ID information and bar code information to a particular advertiser and wand distributor. The ARS 308 returns a message packet over a path 1902 to the user PC 302 which contains routing information as to the location of various other sites disposed on the GCN 306, for example, the advertiser server 312 and wand distributor site 1616.

It is noted that, when the wand (or tool) is utilized, the program at the PC may recognize this and merely input the user ID in the wand (or tool) ID field. Alternatively, the mere receipt of a user ID in association with product code information will trigger the ARS

308 to assume that the wand 1600 was utilized. It is only important that the use of the wand (or tool) be recognized and that the user's PC be routed to a location on the network preassociated with that wand (or tool) distributor. Further, it is the routing of the user's PC to a predetermined location on the network based upon the use of a particular tool, a particular type of tool or a general class of tools that elicits the connection. For example, it would be that the network connection is made in response to the suer utilizing a bar code scanner. This would connect the user to a web site for a general bar code scanner tool. For a disk drive, the program would recognize that a disk drive had been installed (or merely used) and then route the user to the web site of the disk drive manufacturer, a competitor of such or even to some marketing firm that wants to contact individuals that use or initiate such a piece of equipment.

It can be appreciated that other information can also be provided by the ARS 308 which more closely targets the particular user of the wand 1600. For example, if it is known that a particular wand 1600 is sold in a certain geographic area, this information can be useful in targeting the particular user with certain advertising information relevant to that geographic area. In any case, the information returned from the ARS 308 over path 1902 provides enough information for the wand program running on the user PC 302 to identify a number of other sites disposed on the GCN 306. The user PC 302 then processes the return message packet and routes another message packet over a path 1904 to the advertiser server 312. The advertiser server 312 then returns product information of the particular product in which the user was interested back to the user PC 302 over a path 1906. Similarly, the user PC 302 routes information (e.g., the URL of the wand distributor site and the user profile) to the wand distributor site 1616 over a path 1908 in order to obtain information back over a path 1910 for framing any banners which identify the wand distributor. Additionally, the user PC 302 forwards a message packet to the E-commerce site 1618 over a path 1912 in order to return information regarding any particular advertisements the wand distributor wants to display to the user. The advertisements are returned to the PC 302 over a path 1914.

Referring now to FIGURE 20, there is illustrated a block diagram of a browser window according to the disclosed embodiment. The browser window 2000 is partitioned

into a plurality of areas for framing specific information. A bar code area 2002 displays that product information in which the user was interested; a wand specific area 2004 displays information about the wand distributor; and an E-commerce area 2006 displays advertising information that the wand distributor selects for display according to this particular user and wand 1600. As mentioned hereinabove, a program operable to process scanned bar code information with the unique wand 1600 develops the browser window by partitioning it into specific areas for the framing of information. Therefore, information returned from the E-commerce site 1608 is passed through the GCN 306 to the particular E-commerce frame 2006. Similarly, information about the particular product of interest is returned from the advertiser site 312 across the GCN 306 to the particular bar code specific area 2002. Information placed in the wand specific area 2004 is information about the wand distributor which is returned from the wand distributor site 1616 across GCN 306.

Referring now to FIGURE 21, there is illustrated a structure of information contained in the ARS database. The ARS database 310 contains a variety of information required to properly interrogate and assemble packets for obtaining information from the various sites disposed on the GCN 306. The ARS database 310 has a database structure 2100 which contains addresses for the web sites containing the product information requested by the user when scanning the bar code 1606 with the wand 1600. Under a product heading 2102 are listed the particular bar codes and associated routing information for addressing the respective server location. For example, the ARS server 308 may contain any number of advertisers having unique URL addresses associated therewith. Therefore, the bar code 1606 of a particular product is associated with a unique URL address which routes any request for information of that product to that particular advertiser's site. Also part of the ARS database structure 2000 is a heading of wand under which is the wand ID 1804 and the distributor associated with that wand ID 1804.

It can be appreciated that there may be a number of distributors using the disclosed architecture such that each distributor has an ID embedded in the wand which uniquely identifies that wand with the particular distributor. Therefore, the unique wand ID 1804 needs to be listed with the respective distributors of that wand 1600 in order to process the information that needs to be framed and displayed to that particular user. Another heading

under the ARS database structure 2100 is a user heading 2106 which contains profile information associated with that particular user ID 1806. As mentioned hereinabove, the user ID 1806 is obtained via the wand software running on the PC 302 and upon installation or subsequent configuration may request that the user input certain profile information which may be used to target that particular user with products and services which identify with that user profile. The ARS database structure 2100 also contains an E-commerce heading 2108 which contains information related to the bar code 1606 and an advertisement that may be triggered by the request for that information. For example, any bar code 1606 associated with a paper source1600 can be associated with the specific information in the ARS database 310. A user wishing to obtain information about a specific soft drink may, in fact, trigger an advertising response of a competitor product. Similarly, the user interested in information about that particular soft drink may also trigger information which is relevant to that particular product or a product which may normally be served in conjunction with that soft drink. Furthermore, if the user profile indicates that this individual has significant interest in finance or insurance, the request for information regarding this particular bar coded product may trigger advertisement from an E-commerce server 1618 related to information about finance and insurance. It should be noted that the information described as contained within the ARS database structure 2100 is not limited to what has been described, but may comprise any number of pieces of information used to present desired information to the computer display of the user.

Referring now to FIGURE 22, there is illustrated a flowchart of the process of receiving information from the user's perspective, and according to the disclosed embodiment. The wand software running on the user's PC 302 runs in the background until activated by output from the wand 1600. Therefore, flow moves to a decision block 2200 where if a scanned input does not occur, flow moves out the "N" path and loops back to the input of decision block 2200. On the other hand, if scanned input information is received, flow moves out the "Y" path to a function block 2202 where the wand software assembles a message packet containing the bar code information, the wand ID 1804 and the ARS 308 URL address. Additionally, the browser is launched in which this information is placed for transmission to the ARS 308. Flow then moves to a function block 2204 where the browser is partitioned into any number of areas in which information is displayed when obtained

from the wand distributor site 1616, the E-commerce site 1618, and the advertiser server 312. It should be known that although three frames are shown in the particular window 2000 of this embodiment, the number of frames displayed in the window 2000 is limited only by the available real estate of the window 2000 area itself.

After the wand software partitions the browser window into one or more frames in preparation of receipt of return information, flow moves to a decision block 2206 where the computer waits for information to be returned from the various sites disposed on the GCN 306. If information is not returned, flow moves out the "N" path and simply loops back to the input to continue monitoring for receipt of the information. If information has been received, flow moves out the "Y" path to a function block 2208 where routing information for each frame (or partitioned area of the window 2000) is inserted into one or more packets for transmission to the various sites. The various sites then return the requested information back to the PC 302, as indicated in function block 2210. Flow is then to a function block 2212 where the proprietary software working in conjunction with the hosted browser places the returned information into the respective frames of the window. The user, viewing the display at PC 302, then perceives a variety of information, one of which is the particular product information which he or she requested, in addition to wand distributor information, and possibly other advertisements based upon the user's profile.

Referring now to FIGURE 23, there is illustrated a flowchart of the process according to the ARS. The ARS 308 is operable to decode and process message received from the GCN 306. Therefore, flow is to a decision block 2300 where, if bar code information is not received, flow is out the "N" path with loop-back to its input. If bar code information has been received, flow is to a function block 2302 where a matching process occurs to link the bar-coded product information to its respective manufacturer. The ARS database 310 also associates the URL address of the manufacturer's server. When a match is found, the ARS 308 begins to assemble a message packet of information for transmission back to the PC 302, as indicated in function block 2304. The message packet contains the product information and the URL address of the manufacturer's web site. Flow then moves to a decision block 2306 where the wand ID 1804 is compared with the list of wand IDs issued by the particular wand distributor. If the wand ID 1804 is validated, flow moves out

the "Y" path to a function block 2308 where the message packet is appended with the wand ID 1804 and distributor routing address. Flow then moves to a decision block 2310 where the ARS 308 determines if any E-commerce information is to be associated with a particular wand ID 1804. If so, flow is out the "Y" path to a function block 2312 where the message packet is appended with the E-commerce routing string. The E-commerce routing string provides addressing for the E-commerce server 1618. Flow then moves to a function block 2314 where all message packets are returned back to the PC 302 for processing.

Referring back to decision block 2306, if the wand ID 1804 is determined to be invalid, flow moves out the "N" path and jumps forward to the input of decision block 2314, since the lack of a wand ID 1804 interrupts the link to any advertising provided by the E-commerce server 1618. At this point, the only information provided is the link to the adverse server 312 for return of product information. Referring now to decision block 2310, if no E-commerce information is available, flow moves out the "N" path and jumps forward to the input of function block 2314 where the message packet back to the PC 302 contains only the URL of the advertiser server 312, the bar code information, the distributor server 1616 address and wand ID 1804 information.

Referring now to FIGURE 24, there is illustrated a flowchart of the process performed at the E-commerce site. The E-commerce server 1618 receives the message packet from the user PC 302, as indicated in function block 2400, and decodes the packet to perform a match with the bar coded information. Moving on to a decision block 2402, if the match is unsuccessful, flow is out the "N" path to a function block 2404 where the match is rejected. A message may be returned to indicate that a problem occurred and the user may need to re-scan the product bar code 1606. If a successful match occurs, flow moves out the "Y" path to a function block 2406 where the wand ID 1804 is matched with the bar code product information. The bar coded information may be distributed to customers over a large geographic area. However, the wand 1606 may be coded for certain geographic areas. For example, a wand 1600 having an XXX ID may be restricted for sale in the Southwestern United States while a wand 1600 having a YYY ID may be sold only in the Northeast. In this way, geographic areas may be targeted with advertising more appealing to that particular area. Advertising returned to the user PC 302 may be focused further by

obtaining a user profile when the software or wand 1600 are installed. In this way, advertising may be focused based upon the user profile. Therefore, flow moves to a function block 2408 to lookup the E-commerce action based upon the wand ID 1804 and the bar code information. Flow moves to a function block 2410 to assemble all the information into a packet for return to the user PC 302. The product information and/or user profile information may be returned. Flow is then to a function block 2412 where the message packet is transmitted.

Although the preferred embodiment has been described in detail, it should be understood that various changes, substitutions and alterations can be made therein without departing from the spirit and scope of the invention as defined by the appended claims.

WHAT IS CLAIMED IS:

1. A system for launching a web browser on a network, comprising:

a computer having an audio input interface and a communication interface coupled to a computer network;

said audio input coupled to the audio output of a source for receiving an audio signal having encoded therein a unique code that is associated with a predetermined destination on the network; and

a program operable on said computer responsive to receipt and decoding of said audio signal received from said source for interacting with connections of said computer to a web site available on said computer network wherein the location of said predetermined destination is not stored in the computer.

2. The system of Claim 1, wherein said audio input interface comprises:

a circuit for converting said audio output of said source into digital form for processing by said computer.

3. The system of Claim 2, wherein said circuit comprises:

an audio circuit having an input coupled to a microphone and an output; and

an A/D converter coupled to said output wherein an output of said A/D converter is coupled to a system bus of said computer.

4. The system of Claim 1, wherein said communication interface comprises:

a modem for converting digital signals processed by said computer for transmission on said network and for converting signals received from said network into digital form for processing by said computer.

5. The system of Claim 1, wherein said computer network comprises:

a global communications network for interconnecting a plurality of distributed computer systems and private or commercial networks.

6. The system of Claim 1, wherein said audio signal comprises:

a sound effect selected from the group including clapping, clicking, whistling, audible tones, subaudible tones, super audible tones or a combination thereof.

7. The system of Claim 1, wherein said audio output of said source comprises:

a broadcast or recorded program including said unique code encoded in an audio component of said program.

8. The system of Claim 1, wherein said audio output of said source comprises:

a generator of an audible signal for initiating execution by said program in said computer.

9. The system of Claim 1, wherein said unique code comprises:

information selected from the group including product identity, product description, manufacturer identity, advertising messages or program execution commands.

10. The system of Claim 1, wherein said program comprises:

a browser program comprising a hypertext-linked application for selectively accessing information stored in a plurality of databases located at a plurality of web sites on said network;

means for utilizing information encoded in said unique code of said audio signal to create a message packet for linking said computer with a web site on said network operated by a manufacturer or distributor identified by said commercial information; and

means for launching said browser on said computer.

11. The system of Claim 10, wherein said means for launching comprises:

means for launching said browser to said predetermined destination.

12. A method for launching a web browser on a network comprising the steps of:

coupling a computer, having an audio input interface responsive to an audio signal output form an audio source and a communication interface for coupling the computer with a communication network, to an audio source and the network;

receiving the audio signal output, having a unique code that is associated with a predetermined destination on the network encoded therein, at the audio input interface and decoding the unique code for processing by the computer; and

launching the browser program responsive to the receipt and decoding of the audio signal having the encoded unique code wherein the location of the predetermined destination is not stored in the computer.

13. The method of Claim 12, wherein the step of coupling a computer comprises the steps of:

providing an audio input interface for receiving the audio signal output from the audio source;

converting the received audio signal to a form readable by the computer; and

transmitting converted audio signal information to the computer.

14. The method of Claim 13, wherein the step of providing an audio input interface comprises the steps of:

providing an audio circuit having an input coupled to a microphone and an output; and

coupling an A/D converter between said output of aid audio circuit and a system bus of said computer.

15. The method of Claim 12, wherein the step of coupling a computer comprises the steps of:

coupling a modem having a system bus terminal and transmit and receive terminals to a system bus of the computer; and

coupling the transmit and receive terminals of the modem to a telephone line connected to the communication network.

16. The method of Claim 12, wherein the step of coupling a computer comprises the steps of:

providing a connection between the computer and an Internet service provider; and

installing a program for controlling communications of the computer with the Internet communication network.

17. The method of Claim 12, wherein the step of receiving comprises the steps of:

providing a broadcast or recorded program source having the unique code encoded therein;

wherein the unique code includes information selected from the group including product identity, product description, manufacturer identity, advertising messages or program execution commands.

18. The method of Claim 12, wherein said audio signal comprises:

a sound effect selected from the group including clapping, clicking, whistling, audible tones, subaudible tones, superaudible tones or a combination thereof.

19. The method of Claim 13, wherein the step of launching the browser comprises the steps of:

interpreting information in the unique code received during the receiving step; and

launching a browser application for assembling a communication message packet in preparation for communication on the communication network.

20. The method of Claim 19, wherein the step of launching comprises the step of:

launching said browser to the predetermined destination.

21. A system for launching a web browser on a network, comprising:

a computer having an audio input interface and a communication interface coupled to a computer network;

said input interface being coupled to the audio output of a source for receiving an audio signal having encoded therein a unique code that is associated with a predetermined destination on the network; and

means responsive to said audio signal received from said source for launching said web browser on said computer to said destination on said computer network wherein the location of said predetermined destination is not stored in said computer.

22. A method for interfacing a user location on a network to a destination location on the network, comprising the steps of:

scanning a bar code having product information contained therein relating to an associated product, which bar code has no network routing information contained therein;

extracting product information contained within the bar code;

defining routing information over the network from the user location to the destination location in response to the steps of scanning and extracting, which routing information defines the location of the destination location on the network; and

interconnecting the user location to the destination location in accordance with the defined routing information.

23. The method of Claim 22, and further the comprising the steps of receiving information from the destination location in response to interconnecting thereto and displaying the received information.

24. The method of Claim 22, wherein the bar code is disposed on or in close association with the associated product.

25. The method of Claim 24, wherein the bar code comprises a ISBN bar code.

26. The method of Claim 24, wherein the bar code comprises an EAN bar code.

27. The method of Claim 24, wherein the bar code comprises a UPN bar code.

28. The method of Claim 22, wherein the step of defining the routing information comprises defining a universal resource locator (URL) of the destination location over the network in response to the step of scanning and extracting.

29. The method of Claim 22, wherein the step of defining routing information comprises the step of determining the existence of predetermined association between the scanned bar code and routing information for the destination location on the network.

30. The method of Claim 29, wherein the step of determining the predetermined association comprises the step of accessing a database of a plurality of predetermined associations between a plurality of bar codes and associated routing information and determining if there is a corresponding bar code in the database.

31. The method of Claim 30, wherein the step of accessing comprises the steps of:

transmitting the extracted information from the bar code to an intermediate location on the network;

providing an associative database at the intermediate location, which associative database has stored therein the plurality of bar codes and associated routing information;

comparing the extracted information in the bar code received at the intermediate location with the database; and

if there is a corresponding bar code in the database to the extracted information in the bar code, returning the routing information to the user's location.

32. A method for launching a web browser application on a user's computer, comprising the steps of:

providing a browser application on the user's computer that is launchable in response to predetermined browser inputs being received by the user's computer;

providing a non-browser input that is not a portion of the set of predetermined browser inputs;

correlating the non-browser input to simulate one or more of the predetermined browser inputs as a simulated browser input; and

launching the web browser on the user's computer with this simulated browser input to access information on a network.

33. The method of Claim 32, wherein the step of providing a non-browser input comprises:

providing a first portion of the non-browser input that is generated local and external to the user's computer at the user's location;

providing a second portion of the non-browser input that is retrievable from a separate location than the first portion; and

combining the first and second portions to provide the non-browser input.

34. The method of Claim 33, wherein the second portion is not retrieved from a location local to the user's computer.

35. The method of Claim 34, wherein the step of providing the second portion comprises retrieving the second portion from a location on the network by accessing the network from the user's computer to an intermediate location on the network, retrieving the second portion therefrom and transferring the retrieved second portion back to the user's computer for use by the step of combining.

36. The method of Claim 33, wherein the step of providing the first portion comprises reading a machine readable code, the step of reading a machine readable code comprising causing the generation of the first portion.

37. A method for interconnecting a user's location to a destination location on a network, comprising the steps of:

receiving the unique information at the user's location, which unique information has no associated routing information embedded therein;

associating network routing information with the received unique information in response to receipt thereof; and

interconnecting the user's location to the destination location across the network in accordance with the associated routing associated therewith in the step of associating.

38. The method of Claim 37, wherein the network comprises a global communication network.

39. The method of Claim 37, wherein the step of receiving the unique information comprises receiving machine readable code having unique information embedded therein.

40. The method of Claim 39, wherein the step of receiving the machine readable code comprises scanning the machine readable code, decoding the machine readable code and outputting the information encoded within the machine readable code.

41. The method of Claim 39, wherein the machine readable code comprises a product code, which product code is fixedly associated with an associated product.

42. The method of Claim 41, wherein the product code comprises a barcode.

43. The method of Claim 41, wherein the product code comprises an ISBN number associated with printed materials.

44. The method of Claim 41, wherein the product code comprises an EAN barcode.

45. The method of Claim 37, and further comprising the step of receiving from the destination location at the user location display information generated by the destination location which is displayed to the user at the user location.

46. The method of Claim 37, wherein the step of associating comprises:

forwarding the unique information to an intermediate location on the network in response to the step of receiving the unique information;

comparing the received unique information at the intermediate location with a database of routing information, which database of routing information includes a plurality

of associative relationships between predetermined unique information and locations of various destination locations on the network; and

if an association between the received unique information and routing information on any of a plurality of destination locations on the network exists within said database, returning the associated routing information back to the user location effecting a network connection to the destination location indicated by the routing information.

47. The method of Claim 37, wherein the steps of associating and interconnecting include the step of activating a web browser program which facilitates the interconnection over the network in response to receiving the unique information, which web browser program is operable to at least provide the interconnection of the user location to the destination location in accordance with the associated routing information.

48. A method for providing an interconnection relationship between a product and a desired location on a global communications network, comprising the steps of:

disposing a machine readable product code on the product machine readable product code, the machine readable product code having encoded product information contained therein, the product code having no routing information embedded therein which would allow the product code, in and of itself, to cause routing to the desired location over any path on the network;

read the machine readable code;

decoding the product code; and

converting the extracted product code for routing information over the network to the desired location, which routing information defines the manner by which a user or a computer at a user location wherein the machine readable code was read can communicate with the desired location via an interconnection therewith.

49. The method of Claim 48, wherein the product code comprises a UPC.

50. The method of Claim 48, wherein the product code comprises an ISBN.

51. The method of Claim 48, wherein the product code comprises an EAN.

52. The method of Claim 48, wherein the routing information comprises a universal resource locator (URL) that comprises a unique locator on the network to the desired location.

53. The method of Claim 48, wherein the step of converting comprises:

providing a database having stored therein an associative table which relates a plurality of product codes with associated locations on the network, each of the product codes having routing information to that associated location associated therewith; and

comparing the extracted information of the product code with the table in the database to determine the routing information.

54. The method of Claim 53, wherein the database is not disposed at the user's location and is disposed at a remote location on the network, requiring a further step of transmitting the extracting information from the product code to the remote location for comparison therewith and, after determining if there is a match and outputting the associated routing information therefrom, transmitting this routed information back to the user's location.

55. A method for utilizing a product code having product information contained therein for interfacing over a network, comprising the steps of:

extracting the product information from the product code, which product code is disposed on or in close association with an associated product; and

in response to the step of extracting, associating network routing information with the product code information.

56. The method of Claim 55, wherein the product code is machine readable code.

57. The method of Claim 56 wherein the step of extracting comprises scanning the machine readable code to extract the product code information therefrom.

58. The method of Claim 56, wherein the machine readable code comprises a bar code.

59. The method of Claim 58, wherein the bar code represents a UPC.

60. The method of Claim 58, wherein the bar code represents an ISBN number.

61. The method of Claim 58, wherein the bar code represents an EAN number.

62. The method of Claim 55, and further comprising the step of interfacing with the network utilizing the associated network routing information.

63. The method of Claim 55, wherein the step of associating the network routing information with the product code information in response to the step of extracting comprises the steps of:

accessing a remote location on the network from the user's location which the product information is extracted;

transmitting the product code information to the remote location over the network;

comparing the product code information with an associative database having stored therein associations with a plurality of product code information and associated routing information and determining if there is a match;

if there is a match, returning the routing information to the location at which the product code information was extracted.

64. A method for accessing information over a network, comprising the steps of:

utilizing a tool in conjunction with an operation on a user's processor at a user location on the network, the tool having a unique ID; and

in response to utilizing the tool, interconnecting the user's location on the network to a predetermined destination at a remote location on the network, which destination has an association with the unique ID of the tool.

65. The method of Claim 64, wherein the tool comprises a peripheral on the user's processor.

66. The method of Claim 64, wherein the tool comprises a scanner which is interfaced with the user's processor.

67. The method of Claim 66, wherein the scanner is utilized to scan a bar code.

68. The method of Claim 64, wherein the information of the remote location on the network is uniquely associated with the unique ID.

69. The method of Claim 64, wherein the step of interconnecting the user's location on the network to a remote location in response to utilizing the tool having a unique tool ID comprises:

determining the unique tool ID of the tool at the user's processor in response to use of the tool in conjunction with the user's processor;

transmitting the tool ID to an intermediate location on the network in response to determining the tool ID;

comparing the transmitted tool ID at the intermediate location with an associative database, which associative database has contained therein a plurality of remote locations and associated routing information therefore on the network and associations with one or more of the a plurality of unique tool IDs for one or more of a plurality of tools;

if a match is made between the received unique tool ID and the corresponding said unique tool ID in said associative database, then returning the associated routing information from the associated database to the user's location on the network; and

at the user's location on the network effecting a communication link over the network between the user's location and the remote location defined by the returned routing information.

70. The method of Claim 69, wherein the associative database is local to the intermediate location.

71. The method of Claim 70, wherein the tool is used by the user at the user's location.

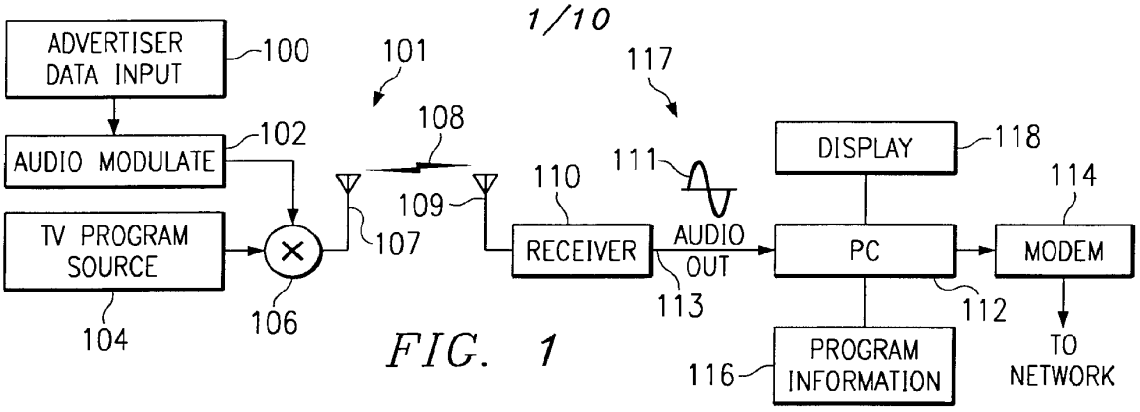


FIG. 1

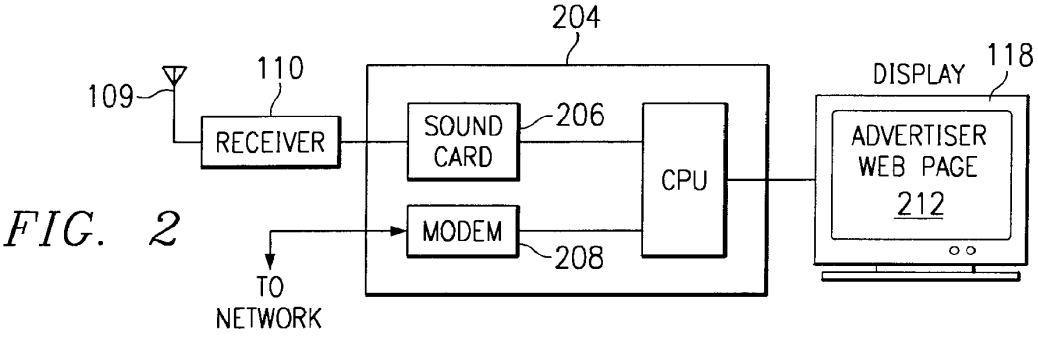


FIG. 2

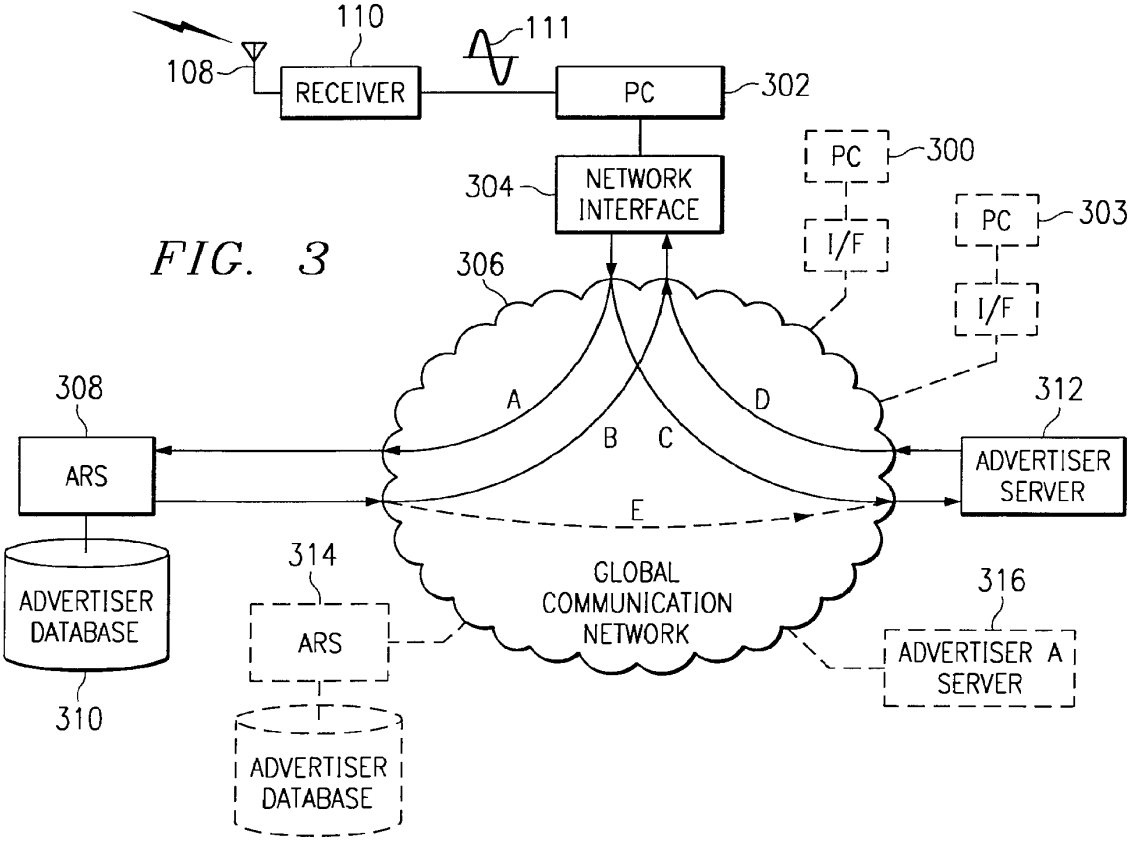


FIG. 3

PATH A: SOURCE TO ARS



FIG. 4a

PATH B: ARS TO SOURCE



FIG. 4b

PATH C: SOURCE TO ADVERTISER



FIG. 4c

PATH D: ADVERTISER TO SOURCE

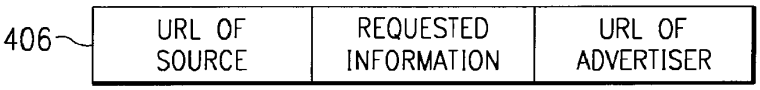


FIG. 4d

PATH E: ARS TO ADVERTISER (OPTIONAL)



FIG. 4e

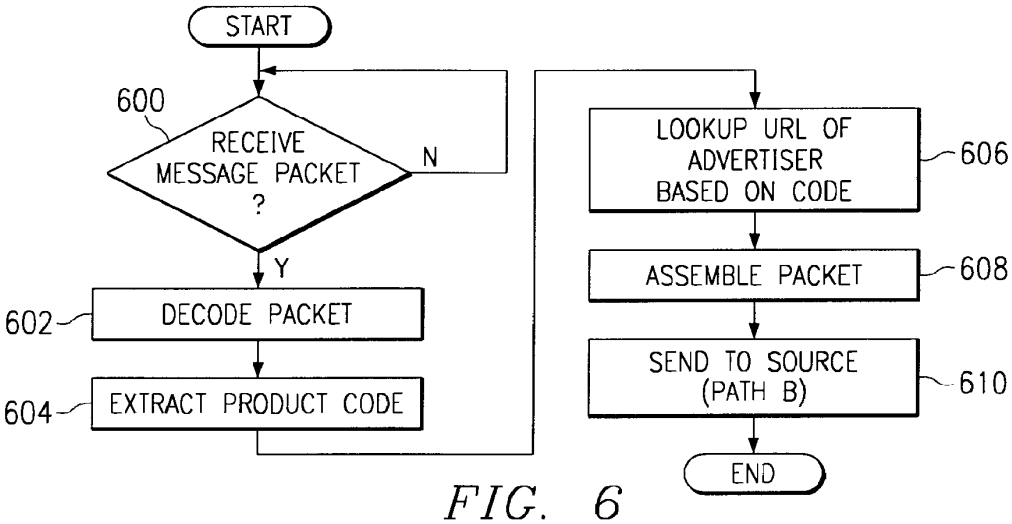


FIG. 6

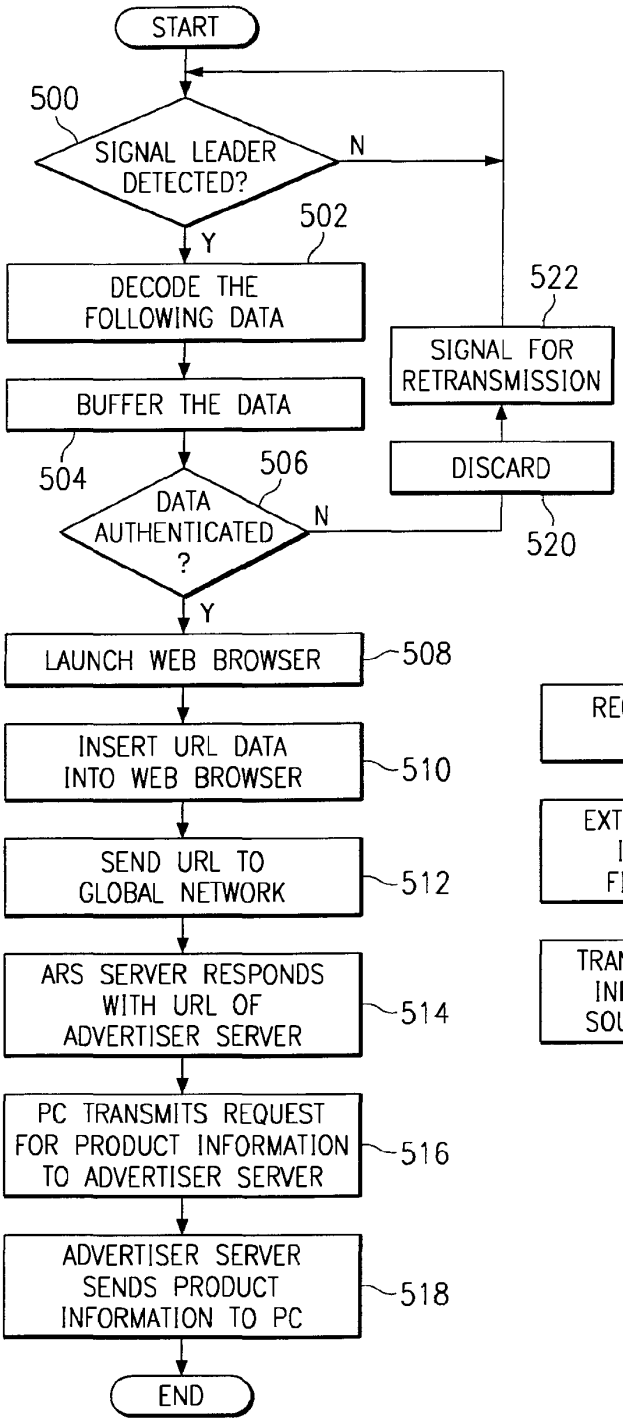

START
500
SIGNAL LEADER DETECTED?
N
Y
502
DECODE THE FOLLOWING DATA
BUFFER THE DATA
504
506
DATA AUTHENTICATED ?
N
Y
522
SIGNAL FOR RETRANSMISSION
DISCARD
520
LAUNCH WEB BROWSER
508
INSERT URL DATA INTO WEB BROWSER
510
SEND URL TO GLOBAL NETWORK
512
ARS SERVER RESPONDS WITH URL OF ADVERTISER SERVER
514
PC TRANSMITS REQUEST FOR PRODUCT INFORMATION TO ADVERTISER SERVER
516
ADVERTISER SERVER SENDS PRODUCT INFORMATION TO PC
518
END

FIG. 5

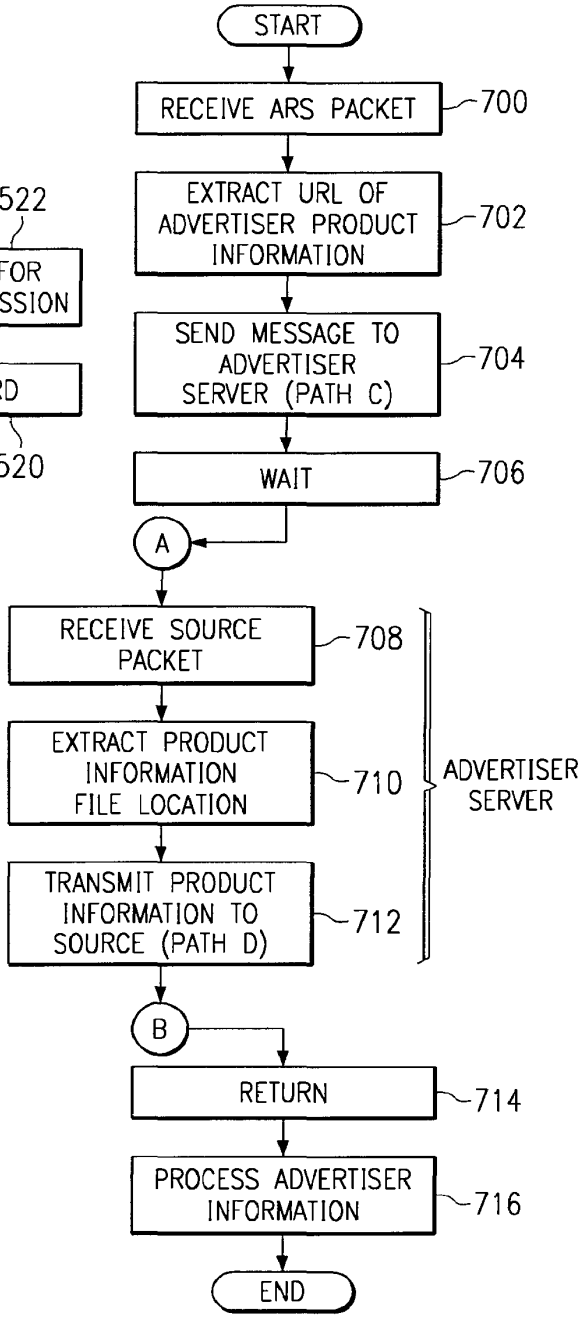

START
RECEIVE ARS PACKET
700
EXTRACT URL OF ADVERTISER PRODUCT INFORMATION
702
SEND MESSAGE TO ADVERTISER SERVER (PATH C)
704
WAIT
706
A
RECEIVE SOURCE PACKET
708
EXTRACT PRODUCT INFORMATION FILE LOCATION
710
TRANSMIT PRODUCT INFORMATION TO SOURCE (PATH D)
712
ADVERTISER SERVER
B
RETURN
714
PROCESS ADVERTISER INFORMATION
716
END

FIG. 7

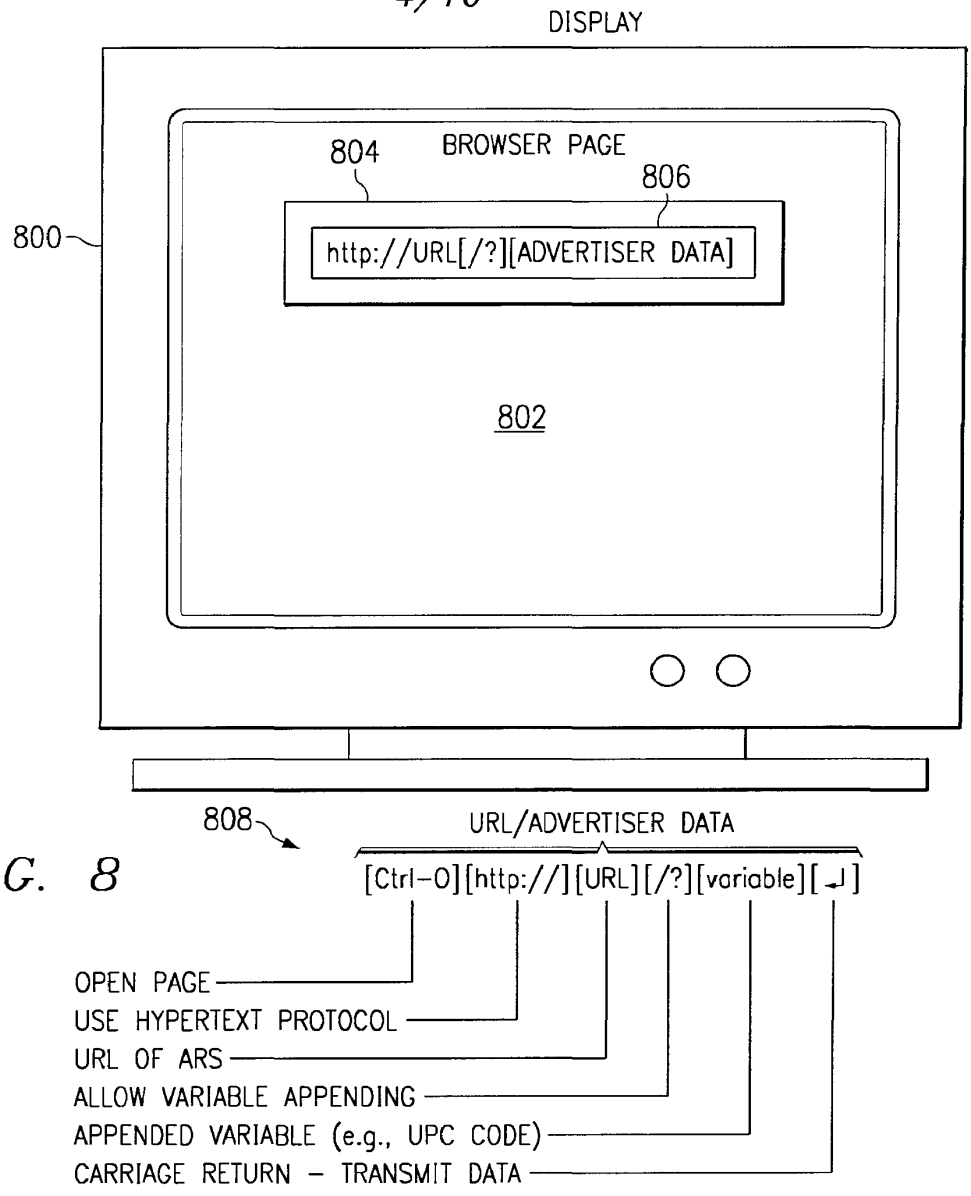

DISPLAY
800
804
BROWSER PAGE
806
http://URL[/?][ADVERTISER DATA]
802
808
URL/ADVERTISER DATA
[Ctrl-O][http://][URL][/?][variable][↵]
OPEN PAGE
USE HYPERTEXT PROTOCOL
URL OF ARS
ALLOW VARIABLE APPENDING
APPENDED VARIABLE (e.g., UPC CODE)
CARRIAGE RETURN – TRANSMIT DATA

FIG. 8

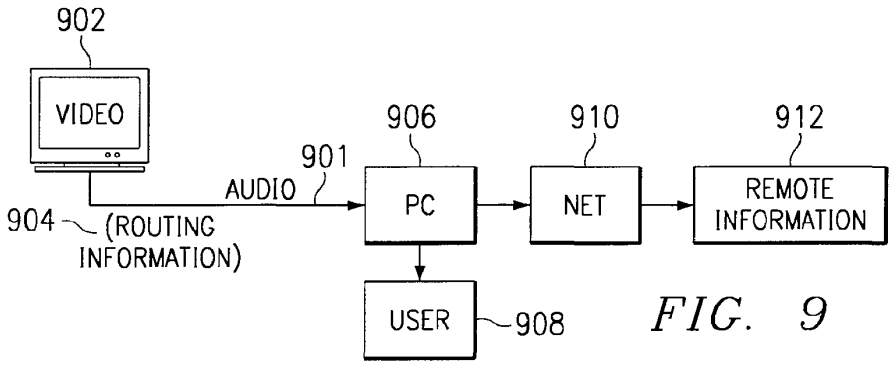

902
VIDEO
901
AUDIO
904
(ROUTING INFORMATION)
906
PC
910
NET
912
REMOTE INFORMATION
USER
908

FIG. 9

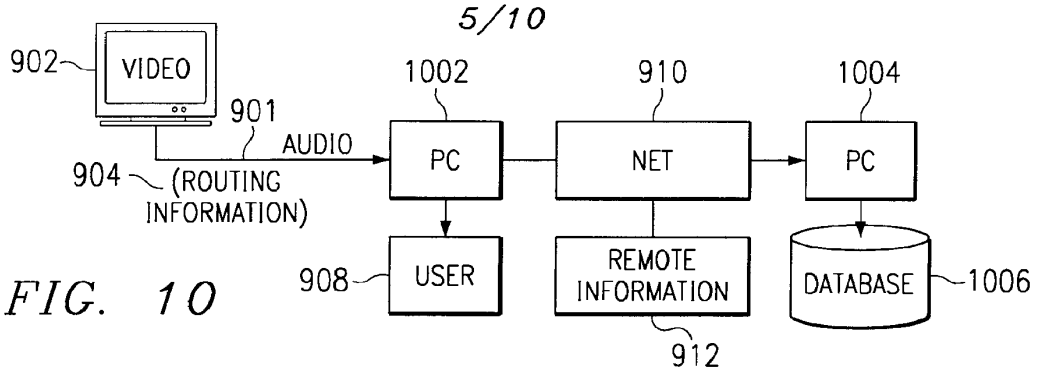


FIG. 10

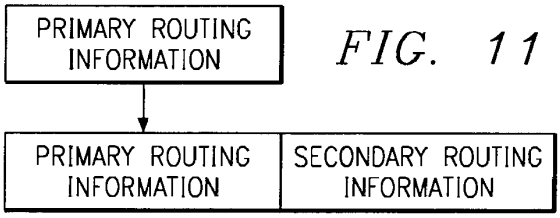


FIG. 11

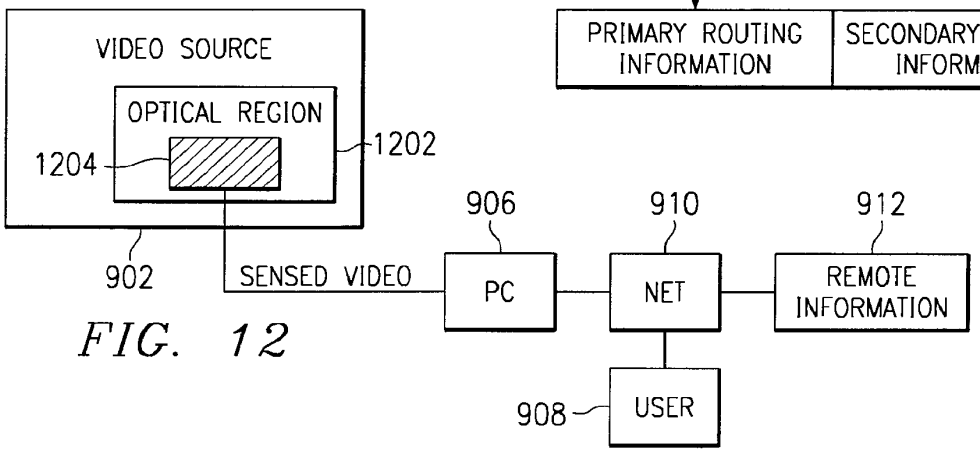


FIG. 12

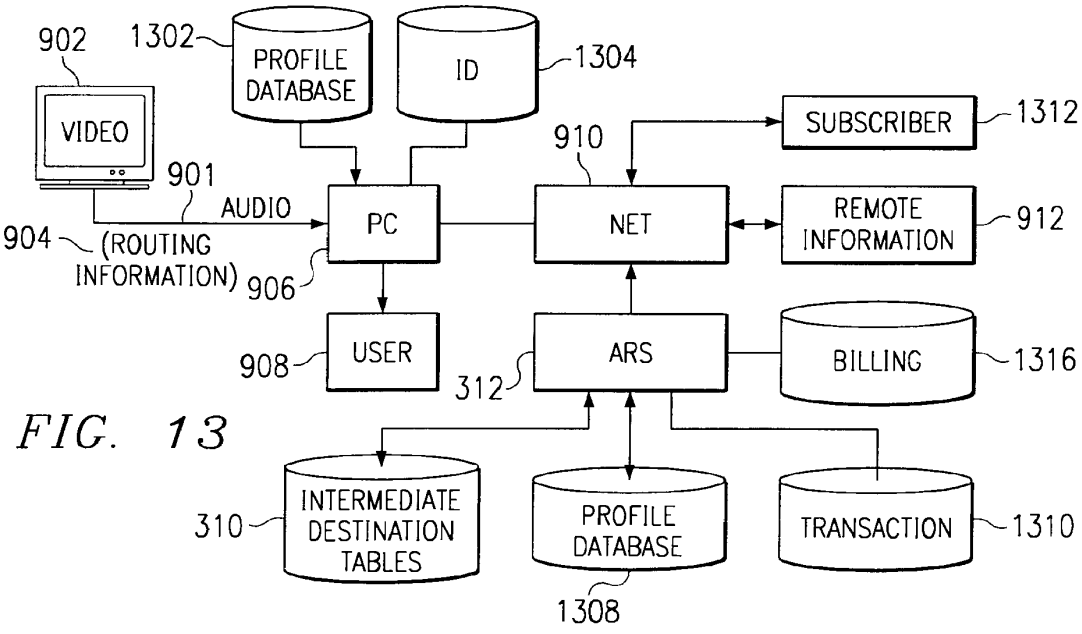


FIG. 13

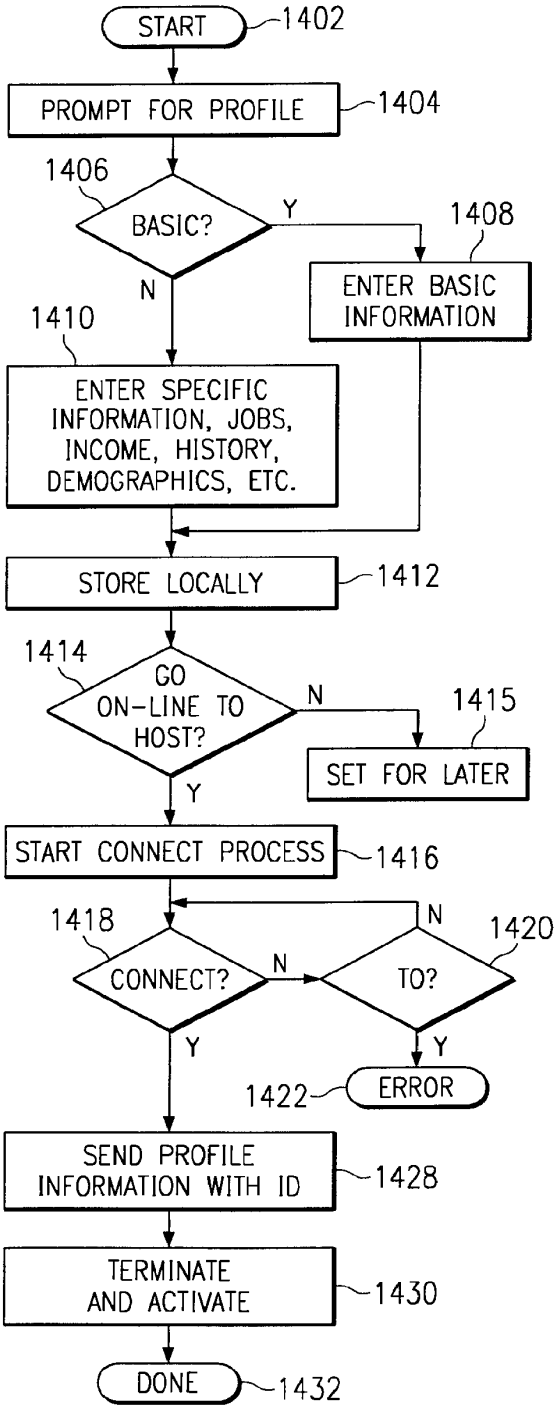

START
1402
PROMPT FOR PROFILE
1404
1406
BASIC?
Y
1408
ENTER BASIC INFORMATION
N
1410
ENTER SPECIFIC INFORMATION, JOBS, INCOME, HISTORY, DEMOGRAPHICS, ETC.
STORE LOCALLY
1412
1414
GO ON-LINE TO HOST?
N
1415
SET FOR LATER
Y
START CONNECT PROCESS
1416
1418
CONNECT?
N
TO?
1420
N
Y
Y
1422
ERROR
SEND PROFILE INFORMATION WITH ID
1428
TERMINATE AND ACTIVATE
1430
DONE
1432

FIG. 14

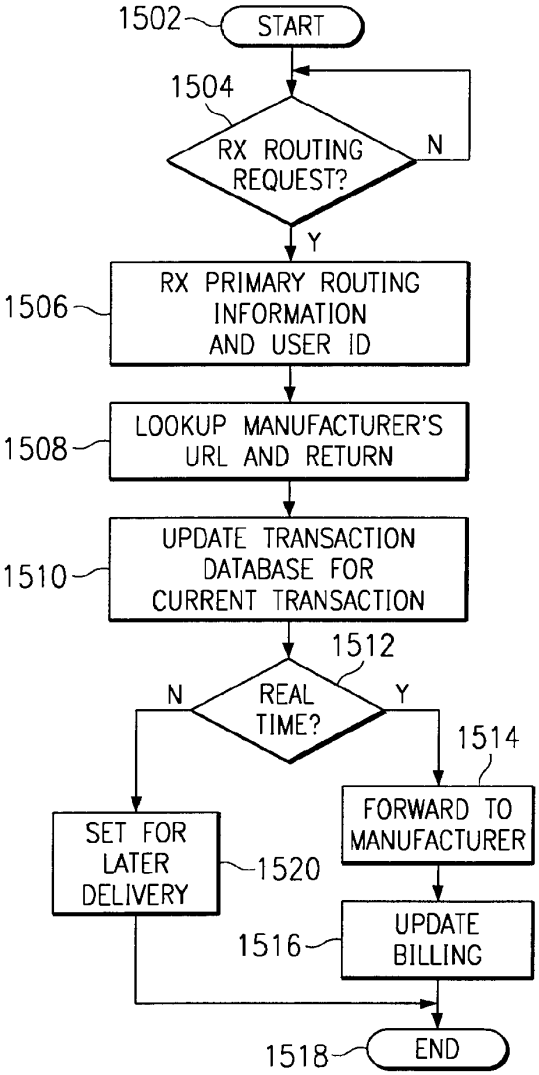

1502
START
1504
RX ROUTING REQUEST?
N
Y
1506
RX PRIMARY ROUTING INFORMATION AND USER ID
1508
LOOKUP MANUFACTURER'S URL AND RETURN
1510
UPDATE TRANSACTION DATABASE FOR CURRENT TRANSACTION
1512
N
REAL TIME?
Y
1514
FORWARD TO MANUFACTURER
SET FOR LATER DELIVERY
1520
1516
UPDATE BILLING
1518
END

FIG. 15

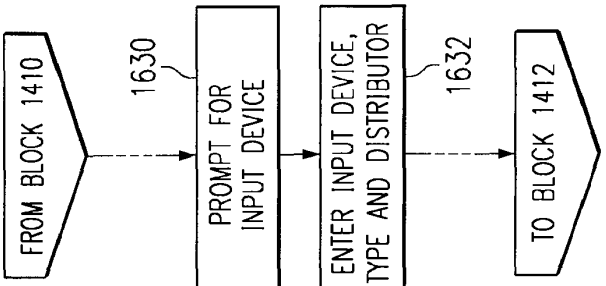


FIG. 16a

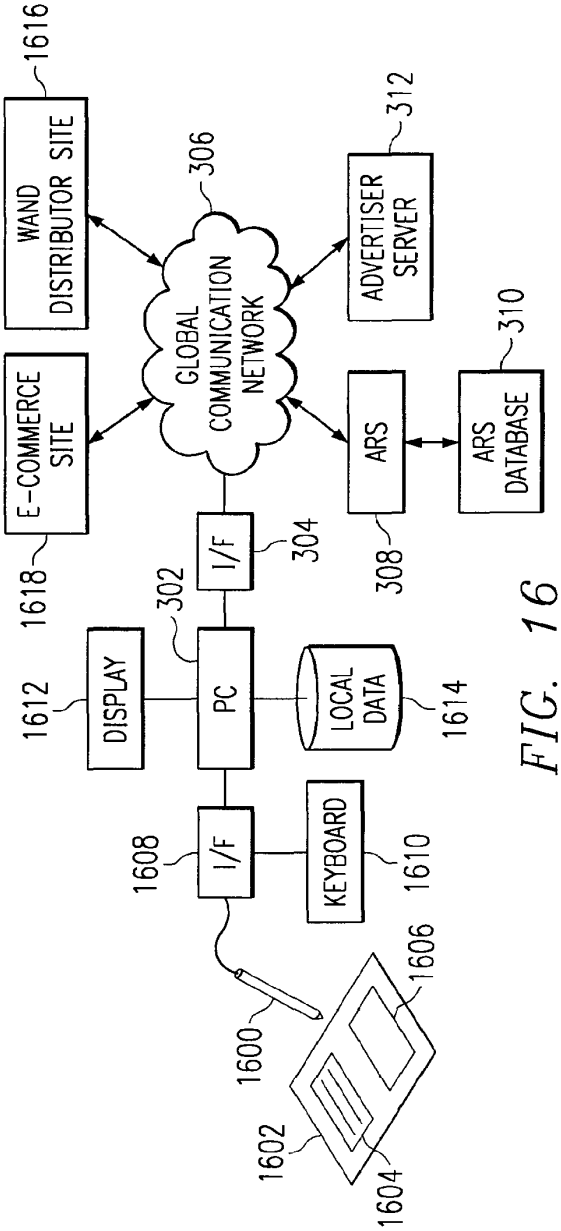


FIG. 16

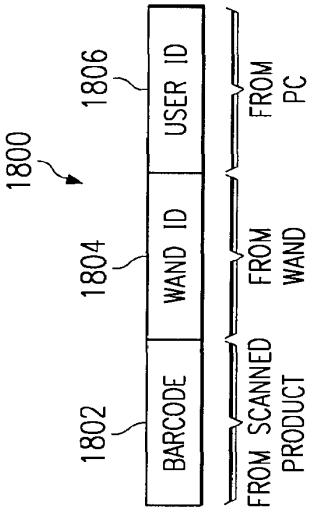


FIG. 18

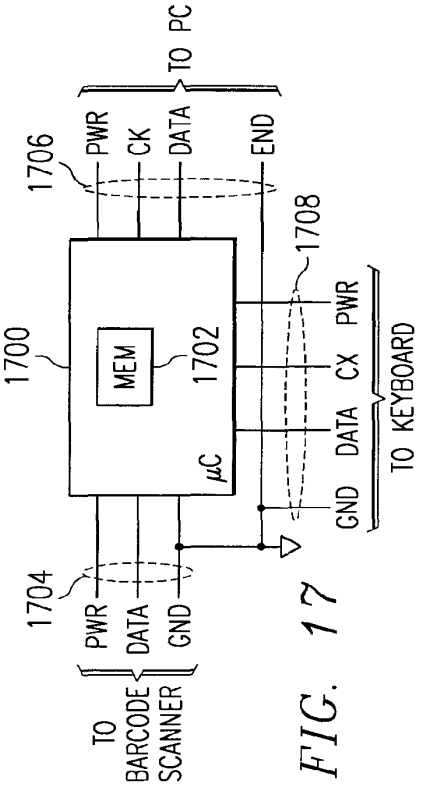
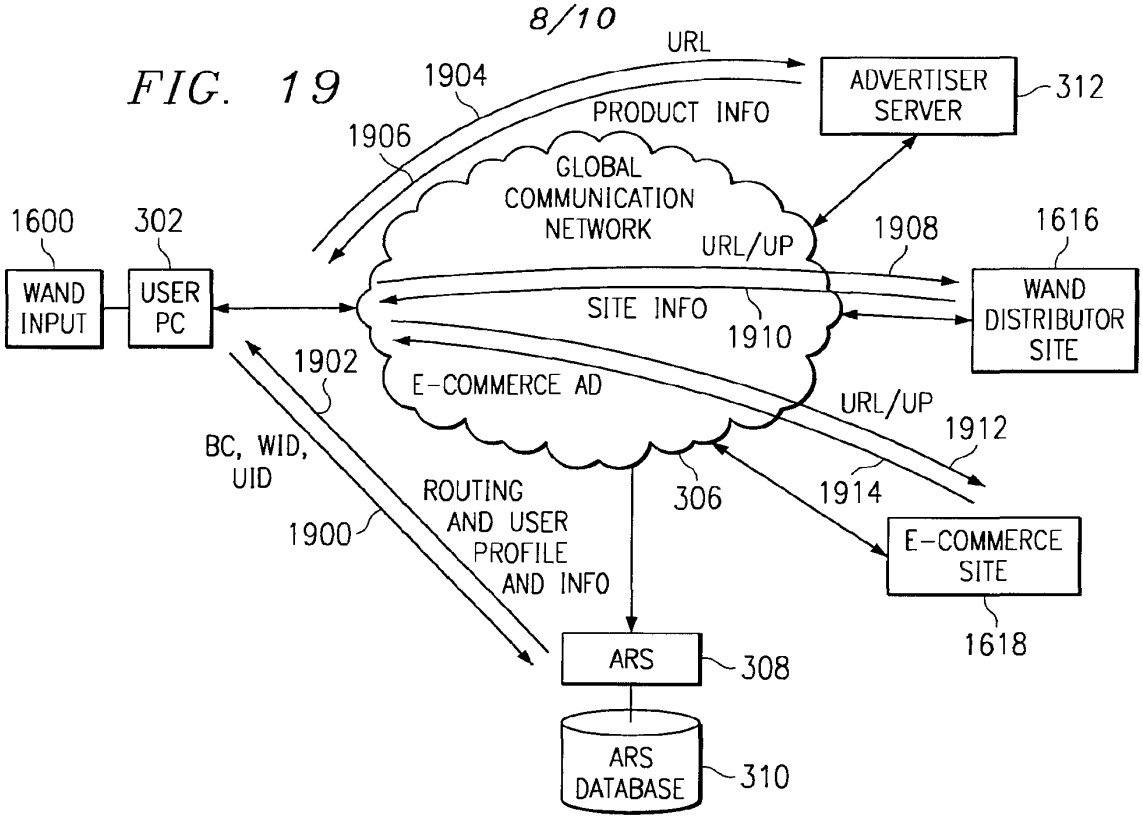


FIG. 17

FIG. 19



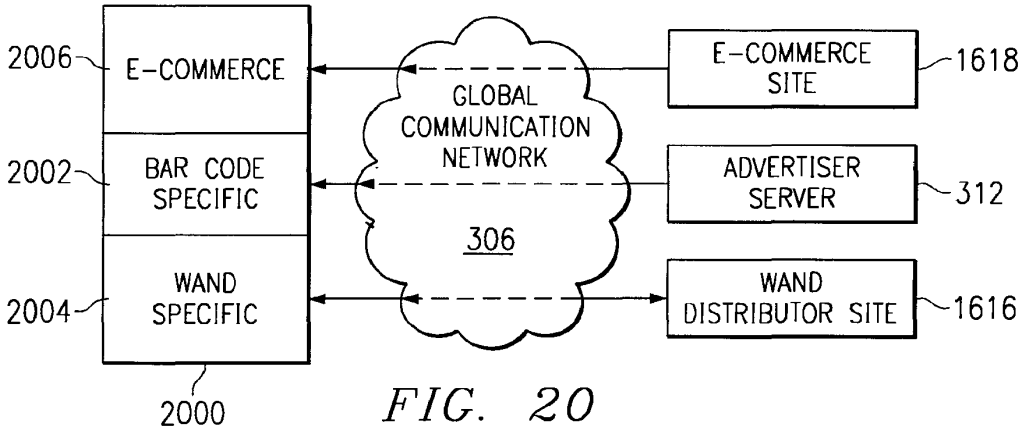


FIG. 20

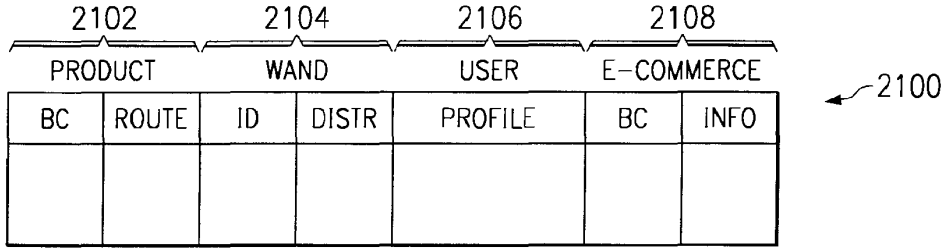


FIG. 21

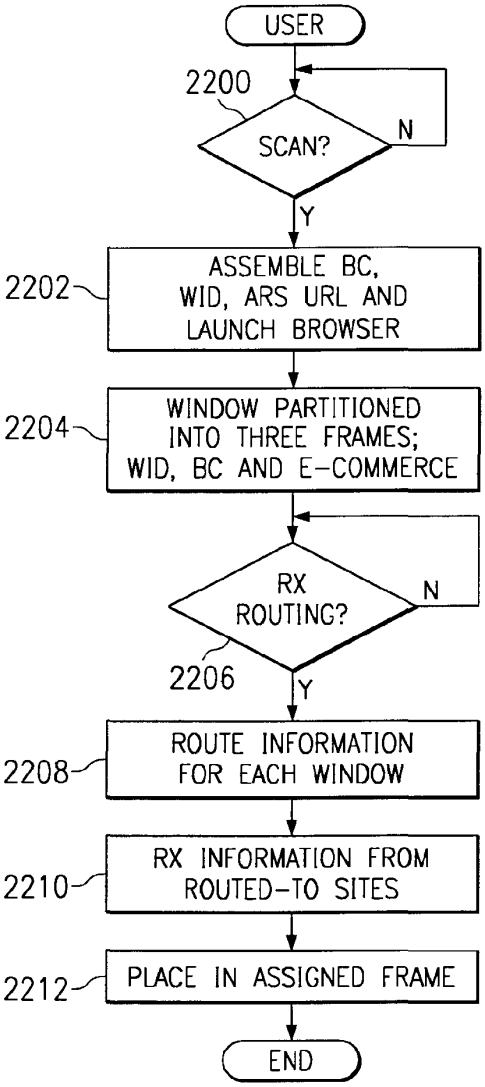

USER
2200
SCAN?
N
Y
2202
ASSEMBLE BC, WID, ARS URL AND LAUNCH BROWSER
2204
WINDOW PARTITIONED INTO THREE FRAMES; WID, BC AND E-COMMERCE
RX ROUTING?
N
2206
Y
2208
ROUTE INFORMATION FOR EACH WINDOW
2210
RX INFORMATION FROM ROUTED-TO SITES
2212
PLACE IN ASSIGNED FRAME
END

FIG. 22

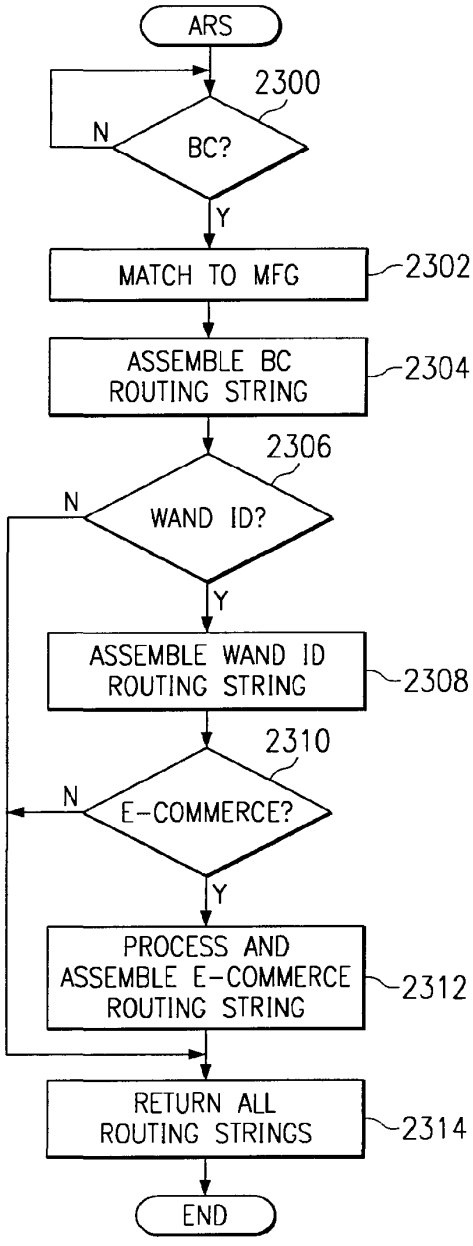

ARS
2300
BC?
N
Y
MATCH TO MFG
2302
ASSEMBLE BC ROUTING STRING
2304
2306
WAND ID?
N
Y
ASSEMBLE WAND ID ROUTING STRING
2308
2310
E-COMMERCE?
N
Y
PROCESS AND ASSEMBLE E-COMMERCE ROUTING STRING
2312
RETURN ALL ROUTING STRINGS
2314
END

FIG. 23

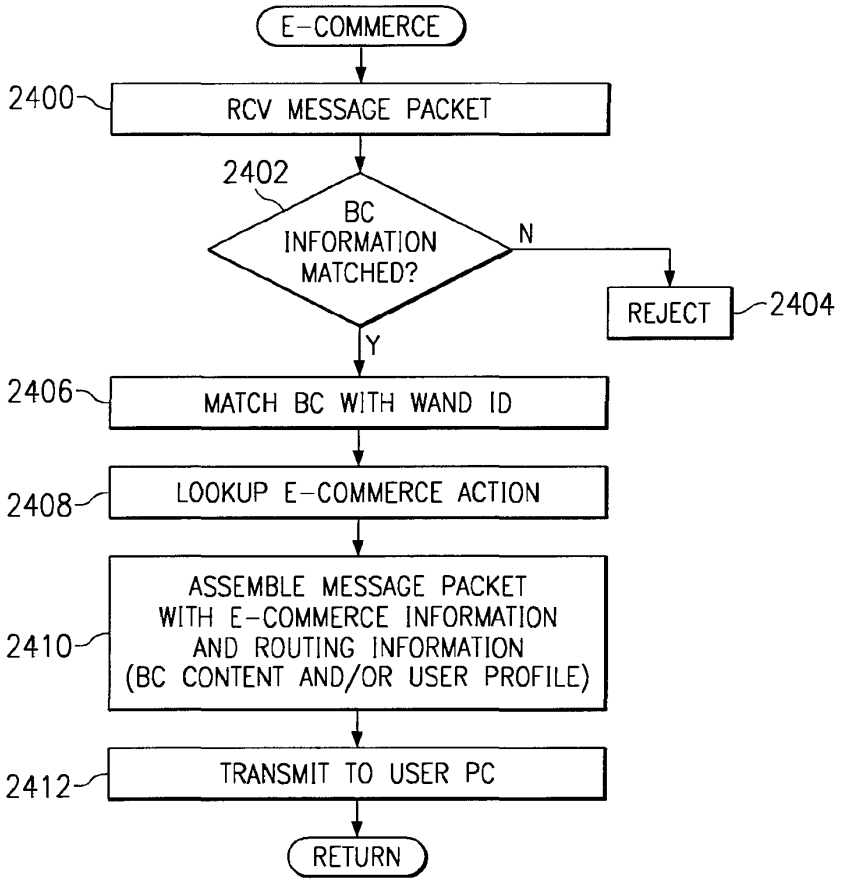

E-COMMERCE
2400
RCV MESSAGE PACKET
2402
BC INFORMATION MATCHED?
N
REJECT
2404
Y
2406
MATCH BC WITH WAND ID
2408
LOOKUP E-COMMERCE ACTION
2410
ASSEMBLE MESSAGE PACKET WITH E-COMMERCE INFORMATION AND ROUTING INFORMATION (BC CONTENT AND/OR USER PROFILE)
2412
TRANSMIT TO USER PC
RETURN

FIG. 24

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
23 August 2001 (23.08.2001)

PCT

(10) International Publication Number
WO 01/61508 A1

(51) International Patent Classification[7]: G06F 13/00, 15/16, H04L 9/00

(21) International Application Number: PCT/US01/04812

(22) International Filing Date: 14 February 2001 (14.02.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/183,681	19 February 2000 (19.02.2000)	US
60/191,778	24 March 2000 (24.03.2000)	US
09/636,102	10 August 2000 (10.08.2000)	US

(71) Applicant *(for all designated States except US)*: **DIGIMARC CORPORATION** [US/US]; 19801 SW 72nd Avenue, Suite 250, Tualatin, OR 97062 (US).

(72) Inventors; and

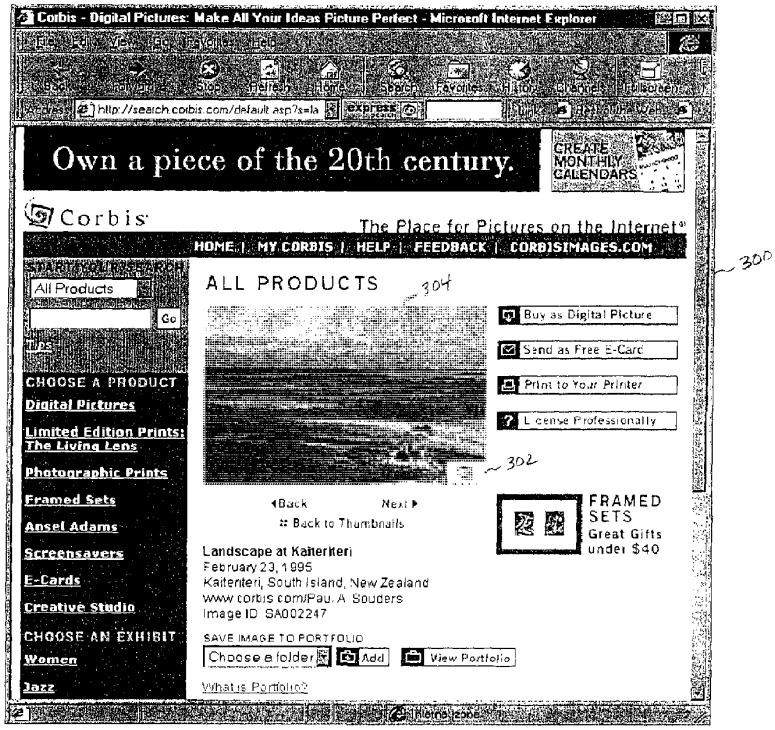
(75) Inventors/Applicants *(for US only)*: **RAMOS, Daniel, O.** [US/US]; 16869 SW Hargis Road, Beaverton, OR 97007 (US). **JONES, Kevin, C.** [US/US]; 4850 NW Neskowin Ave., Portland, OR 97229 (US). **RHOADS, Geoffrey, B.** [US/US]; 2961 SW Turner Road, West Linn, OR 97068 (US).

(74) Agent: **MEYER, Joel, R.**; Digimarc Corporation, 19801 S.W. 72nd Avenue, Suite 250, Tualatin, OR 97062 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

[Continued on next page]

(54) Title: WATERMARK ENCODER AND DECODER ENABLED SOFTWARE AND DEVICES



(57) Abstract: Watermark encoders and decoders are integrated into operating systems, Internet browsers (300), media players, and other applications and devices. Such integration enables the watermark-enabled application (304) or device to provide additional functionality and information (302) available via the watermark. The watermark, for example, may link to metadata or actions related to a media object. To exploit this watermark enabled functionality, the integrated application uses a watermark decoder to access the related metadata and actions. The user interface of the integrated application is enhanced to present metadata and actions linked via the watermark. Similarly, watermark encoders may be integrated into applications to convert media objects into enhanced, watermarked objects.

WO 01/61508 A1

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WATERMARK ENCODER AND DECODER ENABLED SOFTWARE AND DEVICES

Related Application Data

This patent application claims priority to US Provisional Application No. 60/191,778, filed March 24, 2000, which is hereby incorporated by reference. This application also is a continuation in part of applications 09/165,142 filed October 1, 1998, 09/503,881, filed February 14, 2000, 09/507,096, filed February 17, 2000, 09/526,982, filed March 15, 2000, 09/531,076, filed March 20, 2000, and 09/620,019, filed July 20, 2000, which are hereby incorporated by reference.

This patent application is also related to US Patent Applications 09/525,865 entitled Integrating Digital Watermarks into Multimedia Content filed March 15, 2000, 09/563,664 entitled Connected Audio and Other Media Objects filed May 2, 2000, 09/571,422 entitled Methods and Systems for Controlling Computers or Linking to Internet Resources from Physical and Electronic Objects filed May 15, 2000, and 09/574,726, entitled Methods and Systems Employing Digital Watermarking filed May 18, 2000, which are hereby incorporated by reference.

Technical Field

The invention relates to digital watermarking, and specifically relates to applications of digital watermark encoders and decoders in software and devices.

Background and Summary

Digital watermarking is a process for modifying physical or electronic media to embed a machine-readable code into the media. The media may be modified such that the embedded code is imperceptible or nearly imperceptible to the user, yet may be detected through an automated detection process. Most commonly, digital watermarking is applied to media signals such as images, audio signals, and video signals. However, it may also be applied to other types of media objects, including documents (e.g., through line, word or character shifting), software, multi-dimensional graphics models, and surface textures of objects.

Digital watermarking systems typically have two primary components: an encoder that embeds the watermark in a host media signal, and a decoder that detects and reads the embedded watermark from a signal suspected of containing a watermark (a suspect signal). The encoder embeds a watermark by altering the host media signal. The reading component analyzes a suspect signal to detect whether a watermark is present. In applications where the watermark encodes information, the reader extracts this information from the detected watermark.

A great number of particular watermarking techniques are known. The reader is presumed to be familiar with the literature in this field. Particular techniques for embedding and detecting imperceptible watermarks in media signals are detailed in the present assignee's copending application serial number 09/503,881. Other watermarking techniques are known from published patents to NEC (inventor Cox et al), IBM (inventors Morimoto and Braudaway et al), Dice (inventor Cooperman), Philips (inventors Kalker, Linnartz, Talstra, etc. Audio watermarking techniques are known from published patents to Aris (inventor Winograd, Metois, Wolosewicz, etc.), Solana (inventor Lee, Warren, etc.), Dice, AudioTrack, Philips, etc.

This invention relates to integrating watermark encoding and decoding functions in software applications, devices and systems. Watermark encoders and decoders are integrated into operating systems, Internet browsers, media players, and other applications and devices. Such integration enables the watermark-enabled application or device to provide additional functionality and information available via the watermark. The watermark, for example, may link to metadata or actions related to a media object. To exploit this watermark enabled functionality, the integrated application uses a watermark decoder to access the related metadata and actions. The user interface of the integrated application is enhanced to present metadata and actions linked via the watermark. Similarly, watermark encoders may be integrated into applications to convert media objects into enhanced, watermarked objects.

One aspect of the invention is an enhanced file browser system. The file browser enables the user to browse files, including media object files. It includes an extension to decode an object identifier from a selected media object file. The

extension retrieves and displays metadata or actions associated with the media object file via the object identifier.

Another aspect of the invention is a file browser with an extension for encoding an object identifier. This extension enables the user to encode an object identifier into a selected media object file. In an extension to the file browser's user interface, it displays options for controlling the encoding of the object identifier. The object identifier may be encoded into a watermark embedded in the media object.

Another aspect of the invention is a watermark decoder system incorporated into a host application. The host application has a user interface for displaying a representation of media object files. It also has an extension for decoding a watermark from a selected media object file and for displaying metadata associated with the media object file via the watermark.

Another aspect of the invention is an enhanced internet browser. The browser is enhanced with a listener program that identifies media objects in an HTML document, and inserts a handler into the HTML document when it finds an object marked with an object identifier. The handler displays metadata linked via the object identifier in response to user input.

Another aspect of the invention is a method of rendering a media object with branding information. The method decodes an object identifier from the media object, sends the object identifier to a metadata server, receives a brand identifier from the metadata server, and displays a representation of the brand identifier.

Another aspect of the invention is a method for extending a user interface of a media player. In response to input requesting playback of a media object, the method extracts an object identifier from the media object. It uses the object identifier to look up metadata associated with the media object. It then extends the user interface of the media player to include a representation of the metadata associated with the media object.

Further features of the invention will become apparent with reference to the following detailed description and accompanying drawings.

Brief Description of Drawings

Fig. 1 is an example of user interface features enabled by integrating a watermark decoder in an operating system or other application program.

Fig. 2 is an alternative implementation of a user interface for displaying metadata in the file browser of Fig. 1.

Fig. 3 is an example of user interface features enabled by integrating a watermark decoder in an Internet browser.

Fig. 4 shows the example of Fig. 3 with an expanded menu of options linked to a media object via a watermark.

Fig. 5 is an example of user interface features enabled by integrating a watermark decoder in a media player.

Fig. 6 is an example of a user interface features enabled by integrating a watermark encoder in an operating system or other application program.

Fig. 7 is a diagram of a computer system that serves as an operating environment for software implementations of watermark encoder/decoder enabled applications.

Fig. 8 is a diagram with an image containing the words "Confidential".

Fig. 9 is diagram of the fields in a typical digital watermark.

Fig. 10 is a diagram of a typical e-mail system.

Fig. 11 is a more detailed diagram of the watermark reading and detection program shown in Fig. 10.

Fig. 12 is a diagram illustrating a content filtering and indexing system with watermark content filters.

Fig. 13 is a diagram illustrating a distributed watermark detector system.

Detailed Description

Introduction

A watermark may be used to associate a media object such as an image, video or audio file to additional information and actions. The watermark can associate the media object with metadata or processing actions encoded within or stored outside the

media object. Metadata may be encoded in a watermark message, stored within the media object file, or stored outside the media object file. When metadata is stored outside the media object, the watermark may encode an imperceptible and persistent link to this metadata, such as an object identifier (number, address, index, pointer, etc.) within the media object. Wherever the media object travels, watermark decoder-enabled software or devices can extract the watermark from the media signal, and access the metadata or actions associated with the object via the watermark link.

The specific infrastructure for retrieving metadata or actions associated with a media object via its watermark may vary. The metadata or processing actions may reside in a metadata database in the same device or system as the media object, or in a remote device or system accessible via a wire or wireless network. In a distributed computing environment like the Internet, one way is to implement a database server that takes a object identifier extracted from a watermark message and performs one or more tasks associated with the identifier, such as returning metadata or URL links to a requesting computer or device ("client"), routing the identifier to another server, executing some program or set of programs, etc. The watermark message refers to the auxiliary data encoded into a host media object via the watermark.

The following sections summarize ways to take advantage of this functionality in an operating system, Internet browser, and other applications (whether implemented in hardware devices, software, or a combination of hardware and software).

Integrating a Watermark Decoder in Operating Systems and Other Applications

In Operating System

A watermark decoder may be integrated in a file browser component of an operating system. The decoder enables the file browser to extract a watermark and retrieve metadata for watermark enabled media objects. As the user browses files, the watermark decoder may operate as a foreground or background task, automatically or user-initiated, to detect a watermark. Finding a watermark, the browser annotates its representation of the media object as directed by the application(s) associated with the

watermark. Depending on the implementation, the browser may proceed to retrieve additional information or actions associated with the object via the watermark and annotate its representation of the media object accordingly. For example, the browser may annotate the media object with a description of information or actions linked via the watermark (e.g., a brief description and/or http link), or may annotate the object with the actual information or actions.

The process of detecting a watermark and referencing information or actions via the watermark may be implemented to be transparent to the user. For example, the file browser displays information or options obtained via the watermark link without requiring the user to intervene in the watermark detection or information retrieval process.

Alternatively, the file browser can give the user the opportunity to control various stages of watermark detection and processing triggered by the watermark payload. For instance, the user may be given the option to allow the watermark decoder to operate on media objects, and to determine whether and how actions triggered by the watermark payload should proceed. Upon detection of a watermark, for example, the media object can be annotated with an indicator, such as a distinctive sound or logo, that informs the user that information and actions can be accessed via a link embedded in the watermark. The user has the option to access additional information associated with the media object by, for example, selecting a visual logo associated with the object in the user interface. An audio "logo" may be played when the user selects the object (e.g., passes a cursor over its graphical representation in the user interface).

The user interface of the application can be annotated with a variety of graphical and/or audio effects that inform the user of the presence of the watermark link and associated information and actions. Changes in the user interface may be used to convey different stages in the watermark detection and metadata retrieval process. For example, when it first detects the presence of a watermark, the decoder (or host application) plays a generic indicator, such as a simple logo or audio clip. Then, when the appropriate metadata server returns metadata and/or instructions linked via the watermark, the user interface presents specific information associated with the object.

The server may return program code, such as Java Applets, Visual Basic script, XML or some other set of instructions, that present information to the user and provide links to additional information and actions (e.g., URLs or hot links to web sites, other content or program code). Upon receiving this code, a client computer or device executes it. The client typically is the computer or device that decoded the watermark link and issued a request based on the link to the server. This code may perform a variety of functions, including controlling rendering of the watermarked media object and related media objects, some of which may be returned with the linked metadata. The server may return code to control decoding or decrypting the media object or other related media signals to be played along with the watermarked media object. In addition, the server may return code and/or links to enable the user to establish a license and obtain usage rights electronically with a licensing program executing locally and/or on a remote licensing computer (e.g., a licensing server on the Internet).

The server may return data, such as XML, that defines actions to be taken (by providing URLs, instructions, etc.). The client computer or device receiving such a definition of actions may execute the action or present them to the user as options to be executed in response to user input (e.g., clicking on a graphical representation of the option in the user interface of a computer, responding to voice commands via a speech recognition engine, etc.).

The user can access linked information or actions by selecting a graphical representation of the object in a user interface. For example, the user can "click-on" an icon representing the object, or a rendered version of the object (e.g., an image) to determine whether metadata or actions are linked to the object, or to initiate a linked action or retrieval of the metadata.

The watermark decoder may be designed to search for the presence of watermarks in media objects in a specified location (e.g., directories, hard drive, etc.) in response to an event, at periodic intervals, or in response to a user request. For instance, the watermark decoder may be implemented as a utility service, similar to a file search utility, that the user may invoke to a extract watermark link from a media object file, or from a group of files. As another example, the operating system can be designed to invoke the decoder at boot up on all files of a given type in a selected

storage location (e.g., on the hard drive). The operating system may also run the decoder as a background utility to periodically check for watermark links in media objects. A timer or clock service may be used to trigger watermark detection when the timer elapses or at pre-determined time intervals. The operating system may also run the decoder when prescribed events happen, such as downloading a file, saving a file, etc. Triggering the decoder in response to such events enables the operating system to ensure that media objects are checked whenever they enter the system or device in which the operating system is executing, and whenever they are edited.

To improve the efficiency of the watermark decoder, the file system may implement a scheme for tracking when and which media objects have been checked for watermarks. One such scheme is to assign attributes to each media object to indicate whether it has been checked, whether or not it is watermarked, and if so, when the watermark was detected. Then, the decoder uses this information to check only media objects that have not been checked or have been modified, or for media objects for which a given period of time has elapsed since the last check. Each time a media object is modified, the attribute indicating that the mark has been checked may be reset to ensure that only new and modified objects are re-checked.

To illustrate the concept, consider an implementation in the Windows Operating System. Fig. 1 illustrates an example of an extension of the Windows Explorer user interface to support watermark embedding and reading of media objects. Media object files are typically represented as icons 100 in the user interface 102 of the Windows Explorer file browser. By right clicking the mouse while positioning a cursor over the file icon 100, the user can access a context menu 104 of options associated with the selected media object.

This "watermark aware" file browser augments the options of the context menu by listing options such as "Read Watermark" or "Watermark." When the user positions the cursor over the "Read Watermark" option, the operating system invokes a watermark decoder on the media object. The watermark decoder extracts the watermark link and acts in concert with network communication software to access the metadata database and retrieve the items from the database associated with the watermark link. The file browser displays these items in a window 106.

There are a variety of ways to access the metadata database. The metadata may be stored locally (in the same machine as the media object), in a local area network or a wide area network. If the database is located on a remote computer on a computer network, such as the Internet, network communication software may be used to establish a connection with the database.

When selected, the "Watermark" option displays further options for reading and embedding information ("Read Information..." and "Embed Information..."). The window labeled Image Information 106 displays metadata and actions associated with an image via a watermark link embedded in it. In this example, the window 106 displays a thumbnail of the image, image attributes (e.g., width, height, resolution, bits per pixel), and a series of HTML links 108-116 to additional information and actions (such as a search for related images at links 114, 116).

Another way to display items linked via a watermark is to insert them in an additional property page as shown in Fig. 2. While executing the file browser, the user accesses properties (option 118 in Fig. 1) by right clicking the mouse while positioning the cursor over the media object's icon. In response to selecting properties, the operating system displays a properties window such as the one shown in Fig. 2. Each of the property pages associated with the media object has a tab (e.g., General, Image, Watermark, Security). The Watermark page is selected in Fig. 2. Like the options displayed in the window 106 of Fig. 1, the Watermark page lists a series of HTML links 202-210 to additional information or actions. When selected, the link invokes an Internet browser to retrieve information at the underlying URL. For actions like searches 208-210, the link may pass additional parameters such as attributes of the media object to the server at the URL, which in turn, executes a search using those parameters and returns the results to the Internet browser.

An additional way to reflect that a file includes watermarked data is to superimpose a graphical "watermark indicator" icon to the file's icon to signify the presence of a watermark in the file.

While the implementation may vary, the examples depicted in Figs. 1 and 2 are shell extensions of the Windows Explorer file browser. The decoder is implemented as a COM object client of the Windows Explorer. The COM object client is registered

with the operating system as a shell extension handler. Fig. 1 depicts a context menu handler, while Fig. 2 depicts a properties page handler.

Another possible implementation is to implement a shell extension that uses a shell execute command to launch a metadata retrieval application that gets and displays metadata options. This implementation adds an extension to the file browser user interface, such as a context menu extension. When the user selects a media file object within the file browser user interface, it displays a context menu with an option to launch the metadata retrieval program associated with media objects of a given type. A number of actions can be tied to this option. One action is to launch the metadata retrieval application program. Another action is to launch a media player to play the selected option. Of course, both actions can be initiated concurrently in a multitasking operating system.

One example of a metadata retrieval application is a watermark decoder that extracts a watermark message, and forwards an object identifier from the message to a metadata server, which then returns metadata. As noted above, the retrieval application need not extract the object identifier from a watermark if it was already extracted and used recently to retrieve metadata. Instead, the retrieval application can proceed to display the metadata and actions to the user.

To launch the retrieval application, the shell execute command passes the name and location of the media object to the retrieval application. The retrieval application may present its own user interface to display linked metadata and actions, or may pass them to the file browser, which then displays them within an extension such as a properties page or context menu extension. The retrieval application may prompt the user to request permission before decoding a watermark or requesting an update of metadata from the metadata server. Additionally, the retrieval application may launch one or more other applications, such as an Internet browser to issue a request for metadata from a Web server and display metadata and actions in an HTML document returned from the server.

The approaches described above can be implemented for a variety of media object files, including image, video and audio files. Also, the object identifier need not

be inserted in a watermark, but instead may be placed somewhere else in the media object file, such as a file header.

In Browser

The watermark decoder may also be integrated into an Internet browser application. Like the file browser, an Internet browser can browse directories of files in a local computer or across a network. Commercially available browsers like Internet Explorer from Microsoft Corporation and Netscape Navigator typically support transfer protocols like HTTP and FTP, and have additional components for interpreting or compiling code, including HTML, Java, Visual Basic and scripts (e.g., Java scripts, Visual Basic scripts, etc.). In addition, Internet browsers can parse and render HTML for display in a computer or other device's user interface.

To illustrate integrating a watermark decoder in an Internet browser, consider the following example. As the Internet browser downloads and parses web pages with media objects on the Internet, it keeps a listing of these objects, and checks them for the presence of a watermark. In the listing, it annotates the representation of the objects with watermarks to reflect the presence of the watermark, and potentially other data such as a URL where the media object originated. The user may access the listing by viewing an application window (e.g., an application bar) that presents a visual representation of the media objects. For images, one way to represent the media object is through the use of a thumbnail of the image. Images and other objects may be represented as a graphical icon, textual description, or both.

Watermark objects may be distinguished with a visual or audio indicator like a distinctive sound or logo. The user views the metadata or actions associated with a media object by selecting the representation of the watermark-enabled object in some fashion. For example, the user can click on the thumbnail, icon, logo, or textual description to access a menu of metadata or actions linked to the object via the watermark.

Another way to indicate watermarked objects is to alter the appearance of a cursor in a graphical user interface of the software application when the user passes the cursor over the watermarked object or a representation of the object displayed in the

graphical user interface. One way to alter the cursor is to change it from a conventional pointer to a distinctive graphical icon associated with the detected watermarked object type. Another way is to animate the cursor such that it morphs into some other shape or graphical design over a sequence of frames. Similarly, the user interface can alter the appearance of the watermarked object as the user passes the cursor over it. In addition, the user interface can produce a distinctive sound when the user passes the cursor over the object to signify that it includes a watermark.

US Patent Application No. 09/165,142 provides an example of how to decode watermarks from images in HTML pages and annotate the images with a logo indicating that a watermark is present. It describes a way to present a representation of media objects (thumbnails of images) in an application window of a browser. One such application window may be used to display thumbnails of images to present a history of images encountered while browsing web pages on the Internet or elsewhere. The user may click on an image to add it to a separate "bookmark" or "favorites" list. Another application window may be used to display thumbnails of the images in this bookmark list. By selecting an image or a representation of it, the browser links to a network resource associated with the selected image (e.g., via an associated URL). This network resource may be a web page where the image originated. Alternatively, the resource may be a web page referenced via the watermark link embedded in the image. The watermark message may encode a URL or an object identifier that is used to look up a URL of a network resource, such as a web page. For example, the URL might link to the web page of an owner, or to a licensing server.

The methods described in US Patent Application No. 09/165,142 may be applied to other media objects like video and audio signals.

One way to allow the user to access metadata and actions linked to a media object via a watermark is to display them in a menu in the user interface of the Internet browser. Fig. 3 shows an example of how to display metadata and actions associated with a media object in the user interface 300 of an Internet browser. A browser listener program receives events from the Internet browser indicating when a web page has been downloaded. The listener requests from the browser the address or addresses of media objects in the web page. The address indicates where the media object resides

in memory (main memory, virtual memory, cache, etc.). The listener invokes a watermark decoder on the media object or objects, passing it the address of the object in memory.

Finding a watermark, the listener inserts a handler program into the web page in memory. This handler program is responsible for presenting a logo or other indicator to the user indicating the presence of the watermark and providing hot links to information and actions. For example, the indicator in Fig. 3 is a logo 302 superimposed over a rendered version 304 of a watermarked image object in the browser's user interface.

When the user passes the cursor over a logo and selects it, the handler program associated with it displays a menu 400 of options as shown in Fig. 4. The listener program retrieves these options by establishing a connection to a metadata server (e.g., a local or remote database), passing an object identifier extracted from the watermark to the server, and receiving associated items from the server. These items may include URL links to web pages related to the media object, information about the media object, or links to an action, such as licensing server.

While specific implementations may vary, the example depicted in Figs. 3 and 4 is implemented using document view extensions to the Internet Explorer browser from Microsoft Corporation. Microsoft's dynamic HTML provides an interface that allows an Internet Explorer listener program to insert code to modify an HTML document. Using this interface, the listener program inserts a Java script that controls the display and responds to input to the logo superimposed on the image shown in Fig. 3.

In some applications, media content and web site developers may wish to selectively enable or disable functionality associated with a watermark link. One way to implement this feature is to modify the media object file or some other file that acts as a container of a media object (e.g., an HTML) to include a control parameter (like a flag in a header file). This control parameter indicates the presence of a watermark enabled media object and whether the watermark link is enabled or disabled. The control parameter may be designed to default to being active, unless expressly turned off, or vice-versa. When content developers include the object in some multimedia work, such as a website, they can opt to disable the watermark link via the control

parameter. Editing tools and other application programs and devices can be designed to turn the flag on unless expressly instructed to disable the watermark link.

In other applications, it is important that the watermark act as a persistent link to associated metadata. As the media object travels through different systems, gets coded/decoded, gets modified, etc., it may lose conventional metadata stored in the media object file. In these cases, the watermark link may be used to restore the metadata because the watermark is robust to various forms of transformations (e.g., digital to analog conversion, analog to digital conversion, compression, etc.).

Another way to selectively control functionality made available via the watermark is to enable the user to enable or disable watermark decoding on media objects.

The watermark can also be used as an attribute to the browser's file cache system. This allows cache browser applications to identify which cached files are watermarked and perform functions associated with the information embedded in the watermark as described throughout this document.

In Other Applications

The features outlined above can be implemented in any software applications or devices that process media objects. For example, a media object database management system may implement similar functionality. Digital asset management and digital rights management systems may use watermark enabled links to track and control the use of media objects.

Other applications that encounter media objects, like Word processing, spreadsheet, presentation programs, media object editing tools, etc. can all use some version of the features outlined above to link media objects with additional information and actions.

Many application programs have file browser capabilities that can be enhanced using the technology described in this document. For example, the File Open command is often used to browse file objects. The context menu and properties page extensions described above can be implemented for applications that provide file browsing services.

Watermark encoding and decoding functions can also be integrated into file sharing systems, such as the peer to peer file sharing systems like Napster, Gnutella, Freenet, Scour, etc. In a file sharing system, file sharing software executes on a number of client computers interconnected on a computer network. This software tracks the files available for sharing in the computer in which it executes, as well as other computers interconnected via the network. File sharing software may use watermarks or other forms of embedded data in files to control file transfers (uploading and downloading files on the computer), verify that a file is complete and free of viruses, carry metadata that may be used to search for files in the file sharing system, carry links to additional information and opportunities to obtain usage rights or to buy intellectual property rights in the file, or related products or services. To access this functionality, the file sharing software includes watermark or other embedded data decoding software. To insert or alter this functionality, the file sharing software includes watermark or other embedded data encoding software. For more information on this application, see US Patent Application No. 09/620,019, filed July 20, 2000, and entitled Using Embedded Data with File Sharing which is incorporated by reference above.

Context Sensitive Watermark Links

The actions or information linked to a media object via a watermark may be context sensitive. Some examples of "context" include application context (e.g., referring to the application program that is operating on the object), the object location context (where the object resides relative to the user). For example, the behavior of the link may be different when the user is manipulating the object in an editing tool as opposed to inserting the object in a document, such as a word processing document, a spreadsheet, or presentation. Also, the behavior of the watermark link may change based on whether the object is local, in an intranet, or on the Internet, relative to the user.

The decoder application may link to different metadata or processing actions depending on the context of the media object. To implement this functionality, the watermark decoder provides context information to the metadata database, which in

turn, provides context specific metadata or initiates context specific actions. For instance, if the object is being edited with an editing tool, the decoder provides information about the editing tool to the database in addition to the link extracted from the watermark. Similarly, the decoder may provide context information indicating where the object resides relative to the user's computer. Using the context information as a key, the metadata database returns metadata or initiates processing actions that are associated with the context.

Another way to implement the context sensitive behavior is to allow the decoder to control the presentation of watermark-linked actions or information based on the context of the media object. In the case where the context is defined by the application, this approach is akin to giving the application access to all of the linked metadata or actions, and then letting the application control presentation of the linked actions or information to the user. In such a scenario, the metadata database returns at least a descriptor of linked information and actions, and the application chooses which sub-set of the information or actions to apply based on the context. In the case where the context is defined by the location of the object, the decoder operates in a similar fashion. For example, it may choose a sub-set of the linked actions of information based on the location of the object.

A specific example of context information is user information supplied by the user's computer to a web server over the Internet using "cookie" technology. First, the user's computer decodes a watermark from a media object, such as an image, audio, or video signal. It then sends information extracted from the watermark along with a cookie including user information to a metadata server or router (generally referred to as a server). The server operates on user information from a cookie along with information extracted from a watermark in a media object to look up information or actions that are personalized to the user. The metadata server, for example, parses the cookie and uses it to reference information or addresses to network resources (URLs of web pages) in a database that relate to information in the cookie. The server further narrows the pertinent information or links by using the watermark information to look up a subset of information or links that are pertinent to the watermarked object. Then either the server, or another network resource referenced by the database operations

returns associated information back to the users computer for rendering on the display or audio output device. This approach can be used to limit the information returned to specific news, advertising, content, etc. that is likely to be of interest to the user. A similar effect can be achieved by programming the user's computer to supply user preferences that are used in a similar manner as the cookie information.

Supporting Multiple Watermark Types

As watermark technology proliferates, media objects may have different types of watermarks, each associated with a set of watermark encoders and decoders. To accommodate different watermark types, the decoder can be designed to support different watermark protocols. The watermark protocols provide keys and other parameters specifying how to decode a watermark of a given type. Alternatively, a common Application Programming Interface (API) can be specified for different core watermark encoder and decoder software modules or devices. These schemes facilitate the development of many different types of applications and devices that invoke watermark encoder and decoder functions, yet are independent of the watermark protocol and/or core watermark methods.

To support different core watermark methods, the user may install two or more different core watermark encoder/decoder modules. For example, the core modules may be implemented as plug-ins or dynamic link libraries. When installing a module, the installation process updates a registry, such as the registry in the Windows Operating System, to reflect that a watermark type is supported. In this manner, watermark decoders for different media types, and different types of decoders for a single media type may be supported.

In cases where a media object contains a watermark of unknown type, the media object file may specify the watermark type, e.g., through a parameter in a file header. The file browser, or other client of the core watermark module, may invoke the appropriate decoder by extracting the type parameter from the media object and passing it and a reference to the media object to the core module via the API. The API routes the request to the appropriate core module, which in turn, extracts the watermark

message, and returns it to the API. The API passes the message to the requesting application.

In the event that a type parameter is not available, the application or device processing the object may enumerate through all supported watermarking protocols to check if any protocol is present. The watermark protocols for given media or file type may be registered in the device or application (e.g., in a registry of the operating system). To check for each one, the application invokes a watermark screening process for these protocols to determine whether a watermark associated with the protocols is present.

Media Object Branding

The watermark can be used to establish "branding" in addition to facilitating electronic access to other services. In such a branding application, a watermark decoder enabled application or device in a client reads the embedded watermark message from the object and use it to access a digital logo, such as a thumbnail image (e.g., a brand "brand image"). The decoder-enabled application sends an object identifier taken from the watermark message to a server (such as a metadata server on the Internet), which, in response, returns the logo. The application then renders the logo, preferably superimposed on a rendered version of the media object or a graphical representation of it in the user interface of a client computer or device.

In some scenarios, the server also returns an address (e.g., URL) pointing to either a generic "usage rights" server or a custom "usage rights" maintained by the media owner. In an Internet context, the server may return one or more links to related Web sites. Connection rights and corporate branding services may be provided via a central server on the Internet.

Using the object identifier to link to actions and metadata, watermark enabled applications have many options to extend the branding and usage rights services. An application could display "updatable" usage rights associated with the object identifier. In visual media objects, the application may display the branding logo visually superimposed over a portion of the rendered object (e.g., in a corner of a video frame, periodically for a predetermined period of time) or displayed as a splash screen

initiation before playback of the media object begins. To make the branding information less obtrusive, it may be accessible upon request through a menu (e.g., when the user clicks on a representation of the object or "help" menu).

The branding service transforms the notion of copyright notification into a substantial ever-present branding opportunity with additional functionality, such as a hot link to the home page of the content owner or to a licensing server. The branding service may be combined with other watermark enabled functionality, such as a copy management instruction in the watermark payload, e.g., a control parameter indicating whether the object may be played, copied (number of copies allowed), recorded, transferred into other device or system, etc. In addition to providing this instruction to control usage, the watermark payload provides additional value to the consumer (e.g., linking to additional information and services associated with a media object) and ensures that the media object is well labeled and branded during playback.

More elaborate hot branding links, usage rights services, and context-sensitive linking may be added by associating the watermark link with software programs, metadata, and pointers to programs and metadata that support these features. These features may be added in the metadata server by adding them to the list of actions and/or metadata to be executed or returned in response to receiving the media object identifier. In addition, the decoder enabled application may be programmed to send the media object identifier to two or more servers that provide different sets of services or metadata for the object identifier.

Aggregating Metadata

In some applications, a media object may be associated with metadata from two or more different sources. The metadata may be stored in a file that stores the media object. For instance, file formats like JPEG2000, TIFF, JPEG, PSD may allow metadata to be stored along with the media signal (e.g., an image). Metadata and instructions associated with a media object may be stored in the same device as the media object, or in a remote device (e.g., a remote database). In these types of applications, decoders may be programmed to extract metadata from each of the different sources. The media type of file type may signal to the decoder to extract

metadata from these different sources. Alternatively, the watermark message or file metadata may enumerate the different sources of the metadata.

In these applications, the decoder may be designed to get metadata from one or more sources and then present an aggregate of all information. The decoder may perform an aggregation function automatically or prompt the user to select desired sources of metadata for display.

Additional Functionality

A number of features can be implemented that take advantage of the watermark message payload and watermark links to other data and actions. Several examples are highlighted below:

Metadata

Metadata can be expressed in many forms and provide additional functionality. In general, metadata is information about the media object. It may also include machine instructions to be executed, or a reference to information or instructions (object, user, program, or machine identifiers, URLs, pointers, addresses, indices or keys to a database, etc.). The machine instructions may, for example, control rendering, decoding, decrypting or other processing of the object. Alternatively, the instructions may provide some ancillary functionality.

One important application of metadata is to provide ownership information. Another is to provide licensing terms and usage rights.

The metadata can be used to describe attributes of the media object that the user or other applications may use. For example, one attribute may designate the content as restricted, which prevents an application from rendering the content for unauthorized users. Another attribute may designate the object as commercial, which requires an application to seek payment or a license before the object is rendered.

Multiple Watermarks

The media object may contain two or more watermarks or watermark messages, each associated with a distinct set of information or actions. For example, the media object may contain a creator ID, a distributor ID, etc. that link to information about the creator and distributor, respectively.

There are a number of ways to add watermarks to a media object, either at object creation time, or later as the object is transferred, copied, or edited. One way is to interleave separate watermarks in different portions of the media object. This can be accomplished by modifying independent attributes of the media object. Independent, in this context, means that each watermark does not interfere with the detection of the other watermarks. While two independent watermarks may alter the same discrete sample of a media object, they do so in a manner that does not cause an invalid read of any of the watermarks.

Independent watermarks may be located in different spatial or temporal locations of the host media signal (e.g., image, video, audio, graphical model, etc.). The may also be located at different frequency bands or coefficients. Also, they may be made independent by modulating independent features of the signal, such as phase, energy, power, etc. of different portions of the signal.

To illustrate the concept, consider an example of a still image object. Each independent watermark may be defined through a different protocol, which is used to encode a different watermark message (e.g., different watermark links for a creator, distributor, user, etc. of the media object). Independent spatial watermarks may be interleaved by mapping each of the watermarks to a unique set of spatial locations in the image.

In a similar fashion independent watermarks may be encoded in a temporal data sequence, like audio or video, by mapping each watermark to unique temporal locations.

Digital Rights Management

The watermark may be linked to information and processing actions that control use of the media object. For example, the metadata may indicate the owner of the intellectual property rights in the object as well as licensing terms and conditions. Further, the watermark link or metadata trigger processing actions that control use of the object, such as requiring the user to submit payment, and exchanging decoding keys (e.g., decryption, decompression, etc.). While some amount of decoding of the object

may be required to extract the watermark, the remainder of the content may remain encoded and/or encrypted until the user obtains appropriate usage rights.

The digital rights management functionality can be implemented in a licensing or usage rights server, such as the metadata server. This server determines the owner and licensing terms based on the watermark message and executes actions required to authorize use of the object, e.g., electronically receiving payment information from the user, establishing and recording a user's assent to the license, forwarding transaction details to the owner, and returning a usage key to the user. As the user plays or renders the media object, the watermark decoder can send a message to the server to log information about the usage, such as instances of use, machine ID of the player, time of use, etc.

Watermark and Context Information for Usage Control

Watermark decoders can also use context information along with information extracted from a watermark embedded in a media object to control use of that object. For example, the watermark decoder may be programmed to control the rendering, editing, copying or transfer of a media object depending on control data in the watermark and context information derived from the device or system in which the object resides. For example, the watermark may instruct the decoder to inhibit rendering of a media object if its outside of a given file (e.g., a specified web page, computer system, computer network, etc.). After decoding the watermark including such control data, the decoder determines the pertinent context information that must be present to enable a particular operation. This may encompass such actions as verifying the presence of a user identifier, computer identifier, file identifier, storage media identifier, computer identifier, network identifier, etc. before enabling the operation. These identifiers are stored in association with the entities that they identify, or are dynamically derived from these entities. For example, a file identifier can be stored in a file header or footer, or derived from content in the file. A storage device identifier can be stored on the storage device, or derived from content on the storage device or some attribute of it.

This context sensitive control of media objects is particularly useful in controlling the use of media files like music and movies, but applies to other types of

media signals in which watermarks can be encoded. For instance, such context sensitive control can be used to prohibit rendering, copying or transfer of a media object when it is removed from the context of a web page, a computer, a storage device (e.g.., a CD or DVD), file sharing network, computers of paid subscribers of a subscription service, a collection of related media objects, etc.

Asset Management

The watermark embedded in a media object may play a role in asset management. Every time the object is processed (opened, edited, copied, etc.) an application equipped with a decoder can log information about the processing event. Alternatively, it can send a transaction event to a monitoring server via the Internet. The transaction event may specify information about the object, such as its ID, the user's ID, the time and location of use, the nature of the use (number of playbacks). The monitoring server or application records this transaction event in a database and, upon request, generates reports about the use of a given object, or by a given entity. Any of the fields of the transaction record can be used to query the database and generate custom reports.

Integration with Directory Services

Directory services like the Lightweight Directory Access Protocol (LDAP) can use the watermark or operate in conjunction with the watermark decoder to provide additional functionality. LDAP is an on-the-wire protocol used to perform directory operations such as read, search, add, and remove.

For example, an LDAP service can be used to determine when to extract the watermark link and update attributes of a media object. For example, the LDAP service may control periodic updates of the media object's attributes by invoking a watermark decoder and retrieving an update of its attributes from a metadata server at predetermined times. An LDAP search filter that includes or accesses a watermark decoder can also be provided to find watermarked media signals in files stored in file directories on computers.

Media Object Player and Delivery Integration

A watermark decoder may be integrated with software and devices for playing or editing media objects. There are a variety of commercially available players from Liquid Audio, Microsoft, and RealNetworks, to name a few. Such integration allows metadata and actions linked to the media object to be accessible to the user in a seamless fashion through the user interface of the player. For example, when a user is playing a video or audio file, a watermark enabled player may access and display linked metadata or actions automatically or at the user's selection. In either case, the user need not be aware that metadata or actions are obtained from a remote server via an object identifier. Also, such integration enables a content owner to link an object to licensing or assert usage control information or actions at playback time, whenever and wherever the object is played. Such information and actions can be implemented by placing data or instructions in a watermark within the object, or putting an object identifier in the watermark that links to metadata or actions outside the object as detailed throughout this document.

In many cases, it is desirable to access the functionality provided via the watermark transparently to the user. As such, the media player can be enhanced to display items linked via the watermark in a manner that appears to be a natural extension of the user interface. When the user or other program invokes the media player to play an object, it displays metadata or actions provided within or linked via the watermark in an extension of the media player's user interface.

These metadata or actions may be retrieved when the media player is launched or at some other time (e.g., as a background task, in response to user request, when the object is loaded to the computer or device where the media player resides, etc.). In one scenario, the media player invokes a watermark decoder on an object in response to a user's request to play the object. In this scenario, the media player passes the address of the object to the watermark decoder, which in turn, attempts to decode a watermark embedded in it. Upon locating a watermark, the decoder forwards an object identifier extracted from it to a metadata server, which returns metadata or actions associated with the object identifier. During metadata retrieval, the media player proceeds to play

the media object. When the linked metadata arrives, the extension to the media player displays returned metadata or actions.

Fig. 5 illustrates an example of an enhanced version of Microsoft's Windows Media player showing metadata and actions. In this example, the user interface window of the player is expanded in a bottom section 502 to show metadata and links associated with a media object. When the user positions the cursor over the items, "Clip," "Author," or "Copyright, the bottom tab 504 displays a URL associated with that item. By clicking on one of the items, the player invokes an Internet Browser, which sends a query to the resource at the selected URL (e.g., request an HTTP request to download an HTML document).

The watermark decoder may be invoked by another application, which launches the media player. Consider a case where a user is browsing audio or video files, either with a file browser or Internet browser. After finding a desired media object for playback, the user selects the object for playback. For example, the user could select a "Play" option via a context menu of the Windows Explorer, or via an insert in an HTML document as explained above. In the first case, a shell extension invokes the watermark decoder to get the associated metadata and executes a program (e.g., COM object, script, etc.) that runs the media player and displays the additional metadata linked via a watermark. In the second case, a Java script or other insert in the HTML document invokes the decoder and starts a program that runs the media player in a similar fashion.

One way to implement a program to control the Windows Media player is through the use of an Advanced Streaming Redirector (ASX) file. The file contains a script that launches the player and displays the metadata (e.g., URL links and information about the object) linked via the watermark. For information about ASX files and the use of these files to control Windows Media Player, see Microsoft's Developer's Network.

Content Authoring Tools

Media content authoring tools, including web page design tools, may include watermark embedding functionality to embed watermarks into content, such as web

page content. This embedded data then signals watermark decoder enabled devices and software to perform functions associated with the embedded watermark data.

Such tools may also include watermark decoding functionality to enable content developers to use the watermark decoding feature to screen media object for watermarks within the content authoring environment. If the watermark within a media object being edited conveys information (e.g., copyright owner information, licensing terms, etc.), then the authoring tool can convey this information to the user. If the watermark includes usage control information, then the authoring tool can control the use of the media object based on that information. For example, the watermark in a media object may convey an instruction that inhibits the authoring tool from editing the media object, unless the user obtains authorization from a licensing server.

Web Server Integration and Related Applications

Watermark encoding and decoding functions may also be integrated into network server application software to support functionality described in this document as well additional functionality. In particular, watermark encoding and decoding can be integrated into web servers.

Watermark encoders can be integrated into web servers to embed watermarks in media content transferred to, from or through the server. One reason for embedding watermarks at the server is to encode transaction specific information into a media object at the time of its transfer to, from or through the server. The transaction specific information relates to an electronic transaction between the server and some other computer. For example, before downloading or uploading a media object file, the server may embed information about the recipient/sender of the file into the media signal in the file (e.g., image, audio or video file). Since the watermark remains in the signal, information about the sender/recipient in the watermark remains with the media signal in the file through digital to analog –analog to digital conversion, file format changes, etc.

The server may embed a link to information or actions (links to related web sites) in the file that is uniquely tailored to the sender's/recipient's preferences. The preferences may be obtained from the user's computer, such as through popular "cookie" technology commonly used in Internet browsers, or may reside in some other

database that associates a user (a user identifier or an identifier of the user's computer) with the user's preferences (e.g., types of content preferences like news, financial information, sports, e-commerce opportunities, etc.). In this case, the server obtains the user identifier and then queries the database for the associated preferences.

The server may also use the preferences obtained in this manner to control what forms of advertising is returned or linked with the file. For example, the user may request the server to download a desired audio, video, or image file. In response, the server gets the user's preferences and downloads the requested file along with advertising information and web site links that match the user's preferences. The advertising information and links can be referenced by embedding a watermark that includes an address of the information, or that includes an index to a database entry that stores the information and/or links to other information, web sites, etc. The user's computer receiving the file downloaded from the server then renders the file and other related advertising information (e.g., provided in HTML, XML or some other conventional data format) from the server or some other server linked to the file via a watermark in the file.

The server may also embed usage control information into a watermark in a media file based on usage control rights requested by and paid for by the user in an electronic transaction between the server and the user's computer. These usage control rights can then be decoded by other applications and used to control rendering of the file, copying, recording, transfer, etc.

Network servers may also include watermark decoding functionality, such as software for decoding watermarks from media signals in files that are transferred to, from, or through the server. This enables the server to perform the many watermark-enabled functions described or incorporated into this document as well as to provide enhanced functionality. For example, the watermark may include usage control data that the server extracts and acts upon. Examples of usage control data include content rating information (adult content indicators), copy or transfer control information, rendering control information, compression/decompression control information, encryption/decryption control information, links to external information or actions, etc.

After extracting this data from the watermark, the server can modify the file based on the extracted data. For example, the server may compress or encrypt the file in a manner specified in the watermark before transferring the file. The user at the computer receiving the file would then need to have a compatible decompression or decryption program or device to render the media object in the file.

In addition to, or as an alternative to modifying the file based on the extracted watermark data, the server can send related information or instructions to the receiving computer that controls of facilitates usage of the file. For example, if the server determined from the watermark that the content was marked as "adult content", then it could send additional information with the file (e.g., HTTP header information along with a web page including watermarked content) to instruct rendering software, such as the browser, how to render the watermarked content.. The rendering software on the receiving computer can then decide how to render the content. For example, if a child is logged onto the computer receiving the file, then the rendering software can opt not to render content in the file marked as "adult content." As another example, the server may decode a watermark that instructs it to send decryption or decompression keys to the rendering software to enable the receiving computer to decrypt or decompress the content. Public key encryption schemes can be used to perform key exchanges between the sending and receiving computers. As another example, the server may decode a watermark that instructs it to send additional data along with the watermarked file including links to web sites based on information that the server decoded from the watermark.

Content Filtering and Counting

Watermark decoders can be used in computers and computer networks to filter watermarked media objects and to count instances of watermarked media objects. Filtering refers to the use of the watermark decoder to decode watermarks from objects that reside in a particular location and control their use, transfer or rendering in response to control data in the watermark, and optionally, in response to additional context data outside the watermark. These media objects may be temporarily stored at the location of the filter, as in case of a device or computer responsible for transferring the media object. Examples of such systems are e-mail servers, firewalls, routers, or

gateways in computer networks that use watermark decoding to control the transfer of certain media objects to other devices or computers based at least in part on watermarks found in the objects. The media objects may also be stored at the location of a filter on a more permanent basis. For example, the filter may be used to screen media objects that a user downloads to or uploads from a mass storage device such as a hard drive or remote personal library of music, image and movie files on a mass storage device accessible via the Internet. The filter may be used to inhibit downloading or uploading from the mass storage device in response to a watermark in a file being transferred, or alternatively, may be used to control rendering of the file.

Object counting refers to a way of logging the number of times a watermark media object is encountered, either by filtering media objects that pass through a particular device or system like a firewall or e-mail gateway, or by actively searching a network of systems like the Internet and screening for watermarked media objects found and downloaded as a result of the search. The logs maintained by watermark decoding systems can be adapted to include additional information about the object, including information from the watermark, such as an owner, user or transaction identifier, tracer data, and information about the object, such as where it was found, how it was being used, who was using it, etc. Tracer data includes data that is embedded in the file in response to some event, such as detecting unauthorized use, copying or transfer of the file.

The watermark decoder may be further augmented to send the log electronically to another device or computer in response to a specific request or in response to events. For example, the decoder can be programmed to send a report to a central database on another computer when the number of watermarked objects encountered has exceeded a threshold, and/or when certain information is found in a watermark, such as a particular identifier or tracer data that was embedded in the media object in response to detecting an unauthorized use or copying of it. Programmatic rules can be established within the decoder to specify the conditions under which watermarked media objects are filtered and counted, to specify which information is logged, and to specify when the logged information is transmitted to another computer.

The watermark based filtering and counting functions can be implemented in a variety of software applications and devices. Some examples include a network firewall, and other client, server, or peer-to-peer software applications that encounter media objects (such as operating systems, media players, e-mail readers and servers, Internet browsers, file sharing software, file manager software, etc.). One particular use of watermark based filtering, screening and counting is to monitor watermarked content sent in or as an attachment to e-mails sent between computers.

Watermark Based Spiders

Prior patent documents by the assignee of this patent application describe systems and methods of automated searching and watermark screening of media object files on computer networks like the Internet. See U.S. Patent No. 5,862,260, which is hereby incorporated by reference. The software used to perform automated searching and compiling of Internet content or links is sometimes referred to as a web crawler or spider.

As extension of the watermark based information retrieval described in US Patent 5,862,260 and marketed by Digimarc Corporation, watermark decoders can be employed in a distributed fashion to perform watermark screening and counting of watermarked media objects on networks, including the Internet. In particular, watermark decoders can be deployed at a variety of locations on a computer network such as the Internet, including in Internet search engines that screen media objects gathered by each search engine, network firewalls that screen media objects that are encountered at the firewall, in local area networks and databases where spiders do not typically reach, in content filters, etc. Each of these distributed decoders acts as a spider thread that logs watermark information as described in this document and those incorporated by reference. Examples of the types of information include identifiers decoded from watermarks in watermarked media objects, media object counts, addresses of the location of the media objects (where they were found), and other context information (e.g., how the object was being used, who was using it, etc.). The spider threads, in turn, send their logs or reports to a central spider program that compiles them and aggregates the information into fields of a searchable database.

Event Scheduling Based on Embedded Data

Watermark decoding may be used in conjunction with an event scheduler to schedule programmatic events that occur in response to decoding a watermark message of a given type. Throughout this document, there are many instances of triggering actions in response to decoding information, instructions, or links from a watermark message. In some cases, these actions are programmatic actions made by software in response to the watermark, while in other cases, these actions are device actions made by hardware circuitry, such as in the case of usage control of media signals in hardware implementations of audio and video players, recorders, etc.

Rather than taking an action immediately upon decoding a watermark message, an event scheduler can be used to schedule programmatic or device actions to occur at a later event, either at a specified time according to a clock or timer, or in response to a subsequent input, action, etc.

One example of this is to schedule a link to a web site to be activated at later, and perhaps periodic events. For example, a watermark decoder implemented in a browser, operating system, media player or other software application decodes a watermark from a media object that links the object to one or more web sites. In response to decoding the watermark, the decoder schedules programmatic actions to occur at later times. Some examples of these actions include: displaying a window with a link to a specified web site at periodic intervals or in response to a programmatic action like the launching of a browser, media player or other application. This approach can be used to prompt the user to buy a product, such as the media object (a music or video track) or some product depicted in the media object. Using this approach, many actions can be scheduled to occur in response to a single decoding of the watermark.

Integrating a Watermark Encoder in Operating Systems and Other Applications

In some applications, it will be useful to encode a watermark or overlay one or more additional watermarks to perform any of the functions mentioned above (e.g., to track uses, refresh usage rights, add links to additional information and actions, etc.).

Watermark encoding functionality could be added to an operating system, Internet browser, or other applications. For instance, through a drag and drop procedure, a user could embed a watermark in a media object as a means of enabling the various functions outlined above.

As another example, a watermark encoder may be integrated in a file browser, Internet browser, media player, or other application using the same integration techniques outlined above for the decoder. Fig. 1, for example, shows watermark encoder functionality integrated into the Windows Explorer file browser via a shell extension. In particular, the watermark encoder is implemented as a shell extension handler. This handler is registered as a context menu extension in the registry of the Windows Operating system. As an alternative, it could be implemented as a properties page extension handler.

To access the watermark encoder in the Fig. 1 example, the user right clicks on a media object, and selects the context menu option called "Embed Information." In response, the handler displays the window 600 shown in Fig. 6. This window enables the user to enter various Ids (e.g., a creator ID, image ID), which are encoded into an image via a watermark. The user may also set or select attributes of the image object. Finally, the user can control the embedding process by adjusting the durability of the watermark through a scroll bar control. The user can compare the original and watermarked versions of the object by selecting the "Original" and "Watermarked" tabs. When satisfied, the user can save the watermarked image and exit the window (e.g., by selecting close). Metadata and actions may be associated with the image object by forwarding them to the metadata server, which associates them with an object ID.

While the example in Fig. 6 depicts a still image object, a similar approach may be used to embed watermarks in other media objects, such as video and audio objects. To compare marked and unmarked audio or video objects, the shell extension may be designed to launch a media player to play the marked and unmarked objects as desired.

Digital Watermarks as a Gateway and Control Mechanism

The Internet presents security challenges to corporations and others who have confidential information and connections to the Internet or other types of computer

networks. Traditionally, documents containing confidential information are marked with a legend or other visual indicia with words such as "CONFIDENTIAL", "PROPRIETARY", etc. The presence of these marks alert anyone handling such documents that they should only be transferred outside of a company's control under special precautions. It is relatively difficult and unusual for someone to manually send such a document inadvertently to an unauthorized receiver. However, the use of electronic communication services, such as e-mail and file transfers over the Internet, makes malicious and inadvertent transfer trivial.

The Internet, electronic mail, and other network file transfer protocols (FTP, HTTP, etc.) simplify and speed the communications process. Such protocols also make it much easier for someone to inadvertently or accidentally send a confidential document to an unauthorized receiver.

This section describes a system and related methods for using digital watermarks to control the transmission and/or receipt of documents transmitted over computer networks such as the Internet. The system and methods can be used to prevent the accidental dissemination of information to unauthorized receivers. Furthermore, while no security system is fool-proof, the system and methods help guard against the intentional, but unauthorized, dissemination of information to unauthorized receivers.

Most electronically transmitted messages contain text. However, electronic mail systems generally allow images (i.e. pictures) or sound bites to be embedded into and form part of a message. For example, a message can contain a "stamp" with the word "confidential" or a message can contain a sound clip with the word "confidential". An image or sound clip that forms part of an electronic message can carry a digital watermark that can be detected and read by digital watermark reading programs. In addition, text may carry a steganographic information, such as by adding or removing characters or lines according to a hidden pattern that does not change the meaning of the text, such as a pseudorandom pattern created by spread spectrum modulating a binary message with a PN sequence. Such a pattern may be used to add or remove spaces at the ends of lines and paragraphs in a desired pattern. Alternatively,

an image of a text document can carry a watermark by toggling halftone dots on or off according to watermark pattern.

The "payload" or digital data in a digital watermark or steganographic message typically has a number of different fields. One or more of these fields can be dedicated to a flag which indicates that the document or image containing the message is confidential or otherwise classified and that it should only be disseminated in a particular manner.

Typically, e-mail enters a transmission network by way of an e-mail server. Programs that can detect and read watermarks are well known and commercially available. In one implementation of our system, the e-mail server passes each e-mail message through a watermark detection and reading program prior to sending the message out over a network. If the watermark program detects a watermark, it interrogates certain flag bits to determine how the message should be handled. If the watermark reading program finds that a particular flag is set, it can take action such as alerting both the sender and a network administrator. If the watermark program finds no watermark or finds that a particular flag is not set, the message is sent over the network in a conventional manner.

The system can serve as a control mechanism for controlling the dissemination and receipt of electronic messages.

Messages and documents also enter the Internet and other electronic networks received from servers such as Web servers and FTP servers. In a similar fashion a watermark detection program can interrogate documents on servers such as Web and FTP servers and take action such as sounding an alarm if a watermark with a particular flag in a particular state is detected. Such a system may be used to screen files on a web server that have been unintentionally or inadvertently added to a web site. Unauthorized files detected by a programmatic screen of the file directories may be deleted or prevented from being transferred in response to requests for the files via FTP or HTTP requests.

A first embodiment of the system is designed to monitor e-mail messages transmitted over the Internet. This first embodiment is used to prevent the accidental dissemination of confidential e-mail messages to unauthorized users. The first

embodiment prevents the transmission of confidential documents to anyone. Alternate embodiments control the transmission of confidential documents to unauthorized users or for unauthorized purposes. For example, it is easy to add addressees to an e-mail message. Someone may address an e-mail message that includes confidential information to a large group of people without realizing that one of the addressee is not authorized to receive confidential information. The system will prevent such an e-mail from being transmitted to the unauthorized person even though the sender included the address of this person in the list of addressees.

A typical confidential document 10 is represented in Figure 8. The document 10 can either be an e-mail message, or alternatively it may be a document that is attached to an e-mail message. The document 10 includes a confidentiality stamp 11 and lines of text. The confidentiality stamp 11 is an image that has the word "confidential" superimposed over a background that has a variety of lines. That is, the background in image 11 has lines of a the width of which are varied to carry a watermark in accordance with the teachings of US application 09/074,034, filed May 6, 1998 (which corresponds to PCT application PCT/US99/08252), and US application 09/127,503, filed July 31, 1998 (which corresponding to PCT application PCT/US99/14532). The disclosures of the above referenced patent applications are hereby incorporated by reference. Alternatively the background of image 11 or other background portions of the document 10 may comprise a weave or tint pattern that carries a digital watermark signal, such as pseudorandom image pattern formed by spread spectrum modulating a binary message with a PN sequence and converting the modulated sequence to an image of multilevel per pixel values. In still another alternative embodiment instead of having an image 11 embedded in the massage, the message may contain an audio clip with the word confidential. The audio clip would be watermarked using audio watermarking techniques. However, in the first embodiment described herein the, image 11 has both a human readable word "Confidential" and a digital watermark that can be read by a watermark detection and reading program.

The data fields and flags in a typical watermark payload are shown in Figure 9. It should be understood that the fields and flags shown are merely representative and

they can take many alternative forms. The system utilizes one of the flag fields to indicate that a particular document is confidential. The other fields can be used in a conventional manner.

Fig. 10 shows an e-mail system with watermark detecting and reading routines for monitoring and filtering content. A relatively large number of individual user terminal 1301a to 1301e are connected to an e-mail server 1302. Such connections can be by an Ethernet LAN, by dial up modems, or other conventional wire or wireless computer network connection. The e-mail server 1302 has an interface 1303 to the Internet 1304 and it receives and sends messages from the individual users' terminals to the Internet. The e-mail server 1302 itself is conventional, but has a programmatic interface for enabling watermark detecting and reading of messages processed by the server. With the present invention, before the e-mail server 1302 transmits a message from one of the individual User Terminals 1301a to 1301e to the internet, the e-mail server passes the message through a watermark detection and reading program 1305. Both the e-mail message itself and any attached documents are passed through this program. The watermark detection and reading program 1305 determines if a message includes a digital watermark. If a watermark is detected, the confidentiality flag bit is interrogated. If the flag bit is set to "confidential", this embodiment returns the message to the sender. Thus, the first embodiment prohibits any confidential information from being transmitted as part of an e-mail message.

A second embodiment provides for a wider array of alternatives. As shown in Fig. 11, the second embodiment includes a database 1401. The database 1401 includes a list of different groups of potential message senders, a list showing different groups of potential message recipients and a set of possible categories indicated by the setting of the various flags in a message. For example, the senders may fall into three designated sender groups S1, S2 and S3. The potential recipients can fall into three groups designated R1, R2, and R3. The database 1401 and the associated logic 1402 can implement logic rules such as indicated by the following table:

Sender Group	Recipient Group	Flag Conditions	Action
S1	R1	011	Send message
S1	R2	110	Do not sent message notify the administrator
S1	R2	001	Send message, and log fact that S1 sent a message to R2.
S1	R2	101	Return message to sender
S2	R1	011	Send message
S2	R3	110	Do not sent message and notify the system administrator

The system includes a buffer 1403 that buffers messages passing through the server in a FIRST-IN, FIRST OUT fashion. This buffer may store the messages themselves or pointers to their locations within the server's file directory structure. A watermark reading program 1404 reads the message and/or attachments, attempts to detect a digital watermark or embedded steganographic message, and if detected, provides pertinent message payload information to a database interrogation program 1405. A sender/recipient filter program 1402 implements control logic that filters the sender and recipient addresses and maps them to previously assigned groupings. This program provides information for a particular message to the interrogation program 1405, which in response, uses the message payload flags and sender/recipient codes to look up the associated control action associated with the message. The system sends the control action to a message distribution control mechanism 1406 that programmatically instructs the server to take the particular action with a message in the buffer 1403.

In summary, the system considers the message sender, the message recipient and the condition of the flags to determine what action should be taken. A system can include many alternatives in addition to those shown above, or a system might include only a few alternatives. For example, the system could include only a list of addresses which are authorized to receive messages which have a confidentiality flag set to "confidential". Such a system would allow confidential documents to be sent only to selected addresses. Alternatively or in addition, the system could include a list of

individuals authorized to send confidential documents. The system could merely check the sender against this list or alternatively, the system could require that a password be entered when such messages are encountered.

It should be noted that the above table gives only representative rules. In an actual system there could be many more groups and some groups might contain a single individual. Furthermore the table could have many more lines. The table above shows only three flag bits. A system could have more or less flag bits as the needs of the particular system require.

In alternate embodiments, the confidentiality stamp could include a watermark in an image by means other than using line width or halftone dot modulation as described above. The background of the stamp could include an image carrying a digital watermark.

In an alternative embodiment, rather than checking for a digital watermark, the system could check for a text string such as "confidential" and take action in response to locating such a text string. Also, the system could check for a hidden steganographic pattern of spaces at the end of the lines or paragraphs.

While the previously described embodiments apply to e-mail systems, similar precautions could be taken with FTP servers or with Web servers.

Content Filtering and Indexing System Using Digital Watermarks

Fig. 12 is a diagram illustrating a content filtering and indexing system. In this system, a digital watermark embedder embeds digital watermarks into content such as images, video or audio, so that the digital watermark is imperceptible or substantially imperceptible in the content when rendered to end-users (e.g., printed, displayed, played, etc.). This form of embedding involves perceptually adapting the embedded watermark signal to the host media signal based on human perceptual models. The embedder, referred to as a watermark content marker 1502, 1504 is incorporated into content servers, such as web servers, streaming media servers, FTP servers, etc. (1506, 1508) and marks content on those servers. The embedder preferably embeds watermark payloads repeatedly throughout the content, such that different temporal or spatial portions carry the same or different payload relating to the content in which it is

embedded. For example, content type flags may change as the content type changes from advertising to programming, and from one rating to another (G, PG, R, X according to content type).

This content is then sent via a communications network 1510 to other systems, such as web browsers 1512, email readers 1514, media players, etc. Along the communication path, the content may pass through a firewall 1516 and LAN 1518. Digital watermark reader programs, shown as watermark content filters 1520, 1522, and 1524, scan the content for digital watermarks, and if found, decode the message payload of these watermarks. The message payload includes information about the content owner (owner ID), distributor (distributor ID), and/or transaction (transaction ID, receiving machine ID, IP address, sender/recipient address, or user ID). In addition, the payload includes content flags that indicate content type, such as adult content ratings, limited access ratings, etc.

The watermark content filters filter content by controlling its transfer or rendering in response to the content flags. In addition, they send the embedded IDs and content type information to a searchable database 1530 that logs information about each filtering event in the database. The system logs additional event information that is derived from out-of-band data like a file header or the machine where the content is detected, such as network address of detection event, sender and recipient address, user name, etc. The content filters implement a set of rules to control transfer and rendering of content. For example, the content filter for firewalls prevents transfer of content files with certain flags or combinations of flags set in the watermark payload. The firewall may be used to prevent transfer of adult content onto the company network, for example.

The content filters for web browsers, email readers and media players prevent rendering of the content based on the content flags and rules established in the software application where the filter is integrated. For example, user defined rules may be used to block rendering of adult content on computers used by children. These rules are activated when a user logs onto the application, supplying a user ID indicating to the filter that a particular user is using the application and certain rules applicable to that user ID are activated.

A similar approach can be used to filter advertising from content for users that have paid higher subscription fees for content. In this case, content including advertising flags is detected via the watermark and removed before rendering. Similarly, certain types of content can be blocked from being rendered on user's machines where the user's subscription does not authorize playback of content with a particular type of content flags embedded in it. The content filter can be used to enable or block rendering of content based on the user's subscription rights and the flags in the content. This approach is particularly suited for distributed peer to peer models where users may receive content from many different peer computers.

The content filtering system also enables the searchable database to keep a searchable index of content indicating computer addresses (URLs or IP addresses) where copies of the content can be found. As such, the system can be used to create a search engine accessible via the Internet to enable users to search the database by content type, content ID, distributor, or other attributes linked to the content ID and find content items and their corresponding addresses.

Fig. 13 is a diagram illustrating a distributed watermark detector system. In this system, digital watermark detectors (1602, 1604, 1606, 1608) are implemented as spider thread programs that screen content for watermarks, and when found, report information from the watermark payload and information about the event (network address of content, user information, time and date, etc.) to a central spider program 1610 executing on server attached to the Internet 1612. The thread programs are implemented in internal systems 1602, such as content databases on company's LANs, firewalls 1604 on LANs, content filters 1606 within LANs or software applications like browsers or operating systems, and search engines 1608, such as web crawlers that fetch vast amounts of linked web pages and index the pages based on their contents.

The spider program 1610 stores the watermark payload and event information in a database 1612. The spider program also cross references the database information compiled from watermark detection events to additional databases that associate content IDs from the watermark payloads with content attribute information, such as the content title name, owner name, artist, lyrics, URLs of related web sites, etc. This enables users to access the spider program, search for content based on title or some

other attribute, and then cross reference to network addresses where the content can be found.

This system enables deep linking of content on devices across vast and complex networks. The system may be adapted for devices like computers connected to the Internet, music appliances linked on a music file sharing network, set-top boxes and personal video recorders linked on a video file sharing network, etc. In some systems, the nodes of the network where the distributed spiders are located also act as content servers. When a user finds a piece of content via an automated search of the searchable database 1612, the user can send a request to the content server where the content is located and request transfer of the content file from the content server to the user's computer or other device, such as music appliance, set top box, wireless phone, etc.

There are a number of potential enhancements or modifications to the distributed content filtering and spidering systems described above. When filtering content based on the watermark, the watermark payload can include content type flags or an index to content type descriptors stored in a database. In the latter case, the system queries the descriptor database to determine the descriptors that define the nature of the content. Either the content flags or the descriptors can be used by search engines to index content distributed on the network by content type.

To expand the spider's database in the system of Fig. 13, the system may also include programs that perform a automated automated crawl of linked web sites, screen the sites for watermarked content, and then use the content flags in the payload or the content descriptors indexed by the payload to create a searchable index of the content by its type. Users can then search this index to find content matching content type criteria, and get a list of content matching that criteria along with the content's address.

The manner in which the filters control rendering may vary. For example, the watermark filter may prevent playback entirely, scramble the playback, or only scramble an objectionable portion based on the content type flags or descriptors. Since watermarks are repeated throughout frames of the content, the filter can offer fine grained control over which frames of audio or video get scrambled or blocked, or which portions of a still image get blocked. In addition, the filter can selectively control the type of content that gets rendered, for example, by swapping in frames of

content based on user demographic, such as age, location, etc., based on user preferences supplied by the user. In order to swap content, the content delivery mechanism can stream alternative frames in response to the filter's control signals.

The watermark filter may use the watermark payload to link to related information. Initially, the watermark reader links content to a default network resource, such as a default web page. In particular, the watermark payload includes an identifier that indexes a database entry. This entry includes the URL of a default web page. However, the content owner/distributor can dynamically update the database entry to specify a new machine behavior for the identifier, such as a new URL specific to the owner's web site.

Operating Environment for Computer Implementations

Figure 7 illustrates an example of a computer system that serves as an operating environment for software implementations of the watermarking systems described above. The embedder and detector implementations are implemented in C/C++ and are portable to many different computer systems. Fig. 7 generally depicts one such system.

The computer system shown in Fig. 7 includes a computer 1220, including a processing unit 1221, a system memory 1222, and a system bus 1223 that interconnects various system components including the system memory to the processing unit 1221.

The system bus may comprise any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using a bus architecture such as PCI, VESA, Microchannel (MCA), ISA and EISA, to name a few.

The system memory includes read only memory (ROM) 1224 and random access memory (RAM) 1225. A basic input/output system 1226 (BIOS), containing the basic routines that help to transfer information between elements within the computer 1220, such as during start-up, is stored in ROM 1224.

The computer 1220 further includes a hard disk drive 1227, a magnetic disk drive 1228, e.g., to read from or write to a removable disk 1229, and an optical disk drive 1230, e.g., for reading a CD-ROM or DVD disk 1231 or to read from or write to other optical media. The hard disk drive 1227, magnetic disk drive 1228, and optical disk drive 1230 are connected to the system bus 1223 by a hard disk drive interface

1232, a magnetic disk drive interface 1233, and an optical drive interface 1234, respectively. The drives and their associated computer-readable media provide nonvolatile storage of data, data structures, computer-executable instructions (program code such as dynamic link libraries, and executable files), etc. for the computer 1220.

Although the description of computer-readable media above refers to a hard disk, a removable magnetic disk and an optical disk, it can also include other types of media that are readable by a computer, such as magnetic cassettes, flash memory cards, digital video disks, and the like.

A number of program modules may be stored in the drives and RAM 1225, including an operating system 1235, one or more application programs 1236, other program modules 1237, and program data 1238.

A user may enter commands and information into the personal computer 1220 through a keyboard 1240 and pointing device, such as a mouse 1242. Other input devices may include a microphone, sound card, radio or television tuner, joystick, game pad, satellite dish, digital camera, scanner, or the like. A digital camera or scanner 43 may be used to capture the target image for the detection process described above. The camera and scanner are each connected to the computer via a standard interface 44. Currently, there are digital cameras designed to interface with a Universal Serial Bus (USB), Peripheral Component Interconnect (PCI), and parallel port interface. Two emerging standard peripheral interfaces for cameras include USB2 and 1394 (also known as firewire and iLink).

In addition to a camera or scanner, watermarked images or video may be provided from other sources, such as a packaged media devices (e.g., CD, DVD, flash memory, etc), streaming media from a network connection, television tuner, etc. Similarly, watermarked audio may be provided from packaged devices, streaming media, radio tuner, sound cards, etc.

These and other input devices are often connected to the processing unit 1221 through a port interface 1246 that is coupled to the system bus, either directly or indirectly. Examples of such interfaces include a serial port, parallel port, game port or universal serial bus (USB).

A monitor 1247 or other type of display device is also connected to the system bus 1223 via an interface, such as a video adapter 1248. In addition to the monitor, personal computers typically include other peripheral output devices (not shown), such as speakers and printers.

The computer 1220 operates in a networked environment using logical connections to one or more remote computers, such as a remote computer 1249. The remote computer 1249 may be a server, a router, a peer device or other common network node, and typically includes many or all of the elements described relative to the computer 1220, although only a memory storage device 1250 has been illustrated in Figure 7. The logical connections depicted in Figure 7 include a local area network (LAN) 1251 and a wide area network (WAN) 1252. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

When used in a LAN networking environment, the computer 1220 is connected to the local network 1251 through a network interface or adapter 1253. When used in a WAN networking environment, the personal computer 1220 typically includes a modem 1254 or other means for establishing communications over the wide area network 1252, such as the Internet. The modem 1254, which may be internal or external, is connected to the system bus 1223 via the serial port interface 1246.

In a networked environment, program modules depicted relative to the personal computer 1220, or portions of them, may be stored in the remote memory storage device. The processes detailed above can be implemented in a distributed fashion, and as parallel processes. It will be appreciated that the network connections shown are exemplary and that other means of establishing a communications link between the computers may be used.

Extensions to Other Forms of Media Object Linking

The approaches described above can be implemented for a variety of media object files, including image, graphics, video and audio files, or files containing two more different media types. Also, media objects may be linked to their metadata via data structures other than a watermark embedded in the object. For example, the object identifier need not be inserted in a watermark, but instead may be placed somewhere else in the media object file, such as a file header. Such an identifier may be inserted

into the header of coded or compressed files. To extract the identifier, a decoder parses the header and extracts the object identifier. Then, the decoder forwards the identifier to a metadata server, either directly, or by launching another application, such as web browser, to issue the metadata request and output the data and/or interpret code returned from the metadata server.

Concluding Remarks

Having described and illustrated the principles of the technology with reference to specific implementations, it will be recognized that the technology can be implemented in many other, different, forms. To provide a comprehensive disclosure without unduly lengthening the specification, applicants incorporate by reference the patents and patent applications referenced above.

The particular combinations of elements and features in the above-detailed embodiments are exemplary only; the interchanging and substitution of these teachings with other teachings in this and the incorporated-by-reference patents/applications are also contemplated.

In view of the wide variety of embodiments to which the principles of the invention can be applied, it should be recognized that the detailed embodiments are illustrative only and should not be taken as limiting the scope of the invention. Rather, we claim as our invention all such embodiments as may come within the scope and spirit of the following claims, and equivalents thereto.

We Claim:

1. A file browser system comprising:

a file browser for displaying in a user interface a representation of media object files stored in memory; and

a file browser extension for decoding an object identifier from a selected media object file and for displaying in an extension of the user interface metadata or an action associated with the media object file via the object identifier.

2. The file browser system of claim 1 wherein the object identifier is decoded from a watermark embedded in the selected media object file.

3. The file browser system of claim 1 wherein the file browser extension displays the metadata or action in a context menu extension of the user interface of the file browser.

4. The file browser system of claim 1 wherein the file browser displays the metadata or action in a property page extension of the user interface of the file browser.

5. The file browser system of claim 1 wherein the file browser extension forwards the object identifier to a metadata server, and displays metadata or an action returned from the server.

6. The file browser system of claim 5 wherein the file browser extension extracts and displays metadata from the media object file along with metadata returned from the metadata server.

7. The file browser of claim 1 wherein the metadata or action is displayed as a URL link to information or a program associated with the selected media object file.

8. A file browser system comprising:

a file browser for displaying in a user interface media object files stored in memory; and

a file browser extension for encoding an object identifier into a selected media object file and for displaying in an extension of the user interface one or more options for enabling a user to enter input to control the encoding of the object identifier.

9. The method claim 8 wherein the file browser extension comprises a watermark encoder for encoding the object identifier into the selected media object file.

10. A watermark decoder system comprising:

a host application having a user interface for displaying a representation of media object files; and

an extension to the host application for decoding a watermark from a selected media object file and for displaying in an extension of the user interface metadata or an action associated with the media object file via the watermark.

11. An internet browser on a computer readable medium, the browser comprising:

a listener program for identifying a media object in an HTML document; and for inserting a handler into the HTML document when an object identifier is extracted from the media object;

wherein the handler is operable to display metadata linked via the object identifier in response to user input.

12. The internet browser of claim 11 wherein the object identifier is decoded from a watermark embedded in the media object.

13. The internet browser of claim 11 wherein the metadata is retrieved from a metadata server by sending the object identifier to the metadata server.

14. A method of rendering a media object comprising:
decoding an object identifier from the media object;
sending the object identifier to a metadata server;
receiving a brand identifier from the metadata server; and
displaying a representation of the brand identifier.

15. The method of claim 14 wherein the object identifier is decoded from a watermark embedded in the media object.

16. The method of claim 14 wherein the media object is a video or an image, and the representation of the brand identifier is a graphic superimposed on a rendering of the video or image.

17. The method of claim 16 wherein the graphic is a hot link to information or an action associated with the media object.

18. The method of claim 17 wherein selecting the hot link causes retrieval of the information or action from a remote server.

19. A method for extending a user interface of a media player comprising:
in response to input requesting playback of a media object, extracting an object identifier from the media object;
using the object identifier to look up metadata associated with the media object;
extending a user interface of a media player to include a representation of the metadata associated with the media object.

20. The method of claim 19 wherein extracting the object identifier includes decoding the object identifier from a watermark embedded in the media object.

21. An electronic messaging system including a mail server which sends and receives messages, said mail server including a reading program which reads

steganographic data hidden in said messages and which controls distribution of said messages in response to data in said data.

22. A system which includes an e-mail server connected to the Internet means for transmitting messages from individual user to said e-mail server, watermark detecting means for detecting and reading watermarks in e-mail messages before such messages are transmitted from said e-mail server to the Internet, means for preventing the transmission of messages from said e-mail server to the Internet if said watermark detecting means detects a watermark which has an indication that the message including said watermark is confidential.

23. A system for controlling distribution of electronic messages that include confidential information, each electronic message having confidential information including a digital watermark carrying data that indicates that the message is confidential, a server which transmits and receives messages, said server including a watermark reading program which reads watermarks in messages and controls the distribution of such messages in accordance with the data carried by any watermarks in the messages.

24. A method of controlling the distribution of electronic messages that include confidential information, said messages including digital watermarks which carry data indicating that the message includes confidential information, reading watermarks in messages prior to transmission of said messages, and controlling the distribution of each electronic message which includes a watermark in response to the data carried by the watermark in the message.

25. A method of transmitting electronic messages from a sender to a receiver which comprises, detecting and reading digital watermarks embedded in such messages to determine how the flags in such watermarks are set,

interrogating a database to determine what action should be taken with a message based upon the identity of the sender, the identity of the receiver and the flag settings in the watermark in the message.

26. A content filtering system in which watermark content filters are distributed in programs executing on computers interconnected via a communications network, the watermark content filters being operable to detect digital watermarks in content entering a location in the network, to read watermark payload information, and to take action to control rendering or transfer of the content based on content type flags in the watermark payload information.

27. The system of claim 26 wherein the content filters report detection events to a searchable database enabling users to search for content and find a network address indicating where the content has been detected by the content filters.

28. The system of claim 26 wherein the content filters execute rules that control rendering or transfer of content based on a user ID and content type flags in the digital watermarks.

29. The system of claim 26 wherein content type flags are used in combination with rules relating to a user's access rights to enable or disable rendering of advertising along with rendering of the content.

30. A distributed watermark spider system comprising:

digital watermark detectors implemented in spider thread programs executing in devices distributed on a communications network; the spider thread programs operable to screen content at corresponding devices for watermarks, and when found, report information from watermark payloads encoded into the watermark and information about the event to a spider program executing on device connected to the network.

31. The system of claim 30 wherein the information about the event includes a network address of detected content enabling remote devices to access the detected content.

32. The system of claim 31 wherein the spider program is in communication with one or more searchable databases storing the watermark payload information, information about the events, and information about the content cross referenced to the content, wherein the searchable database enables users to search for content by name and find related information about the content, including an address where the information can be fetched via the system.

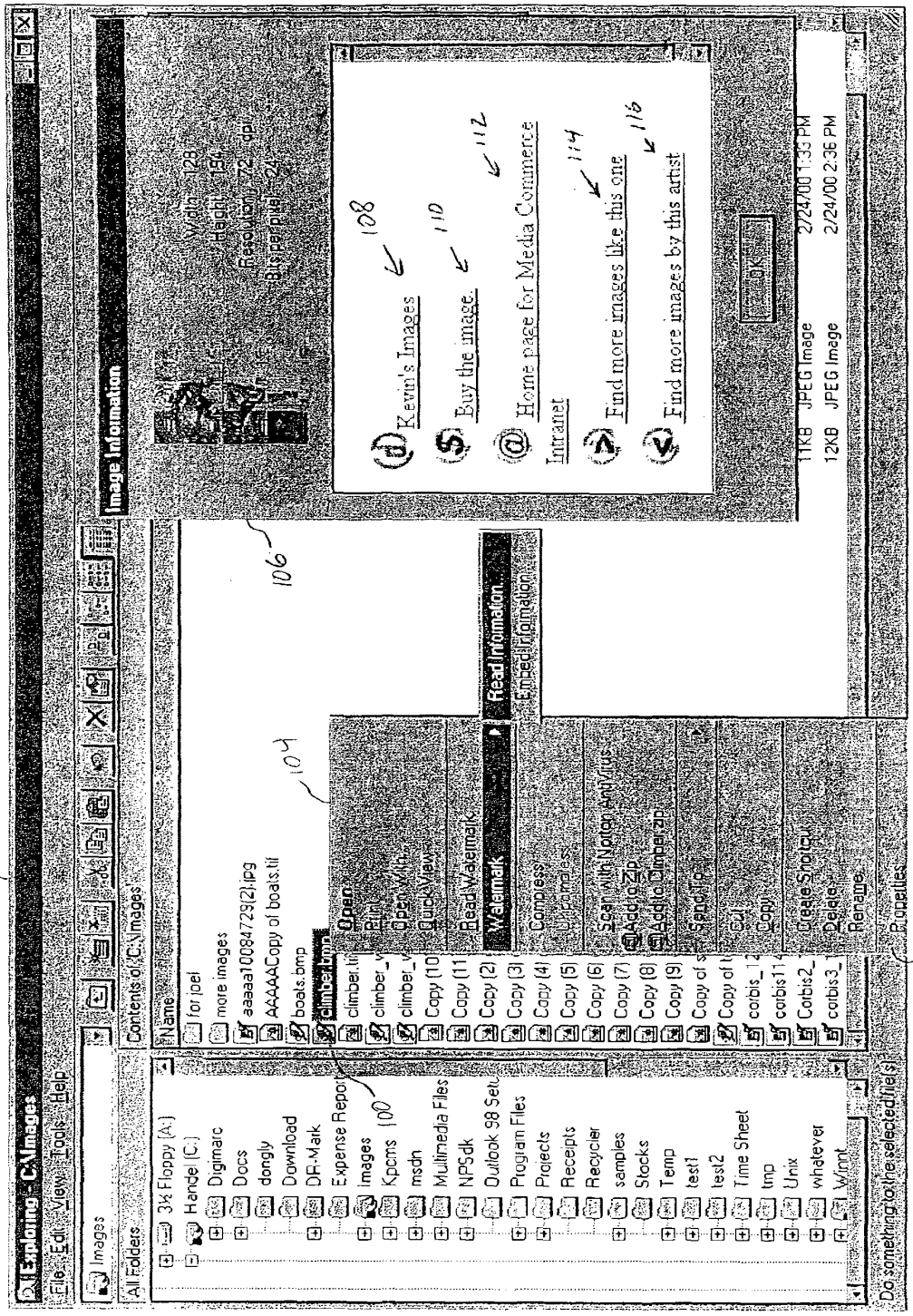

Image Information
Width: 128
Height: 154
Resolution: 72 dpi
BitsPerPixel: 24
Kevin's Images
Buy the image
Home page for Media Commerce Intranet
Find more images like this one
Find more images by this artist
11KB JPEG Image 2/24/00 1:33 PM
12KB JPEG Image 2/24/00 2:36 PM
Watermark
Read Information
Embed Information
Open
Read Watermark
Compress
Uncompress
Scan with Norton AntiVirus
Add to Zip
Add to Climber.zip
Send To
Cut
Copy
Create Shortcut
Delete
Rename
Properties
Do something to the selected file(s)
100
102
104
106
108
110
112
114
116
118

Fig. 1

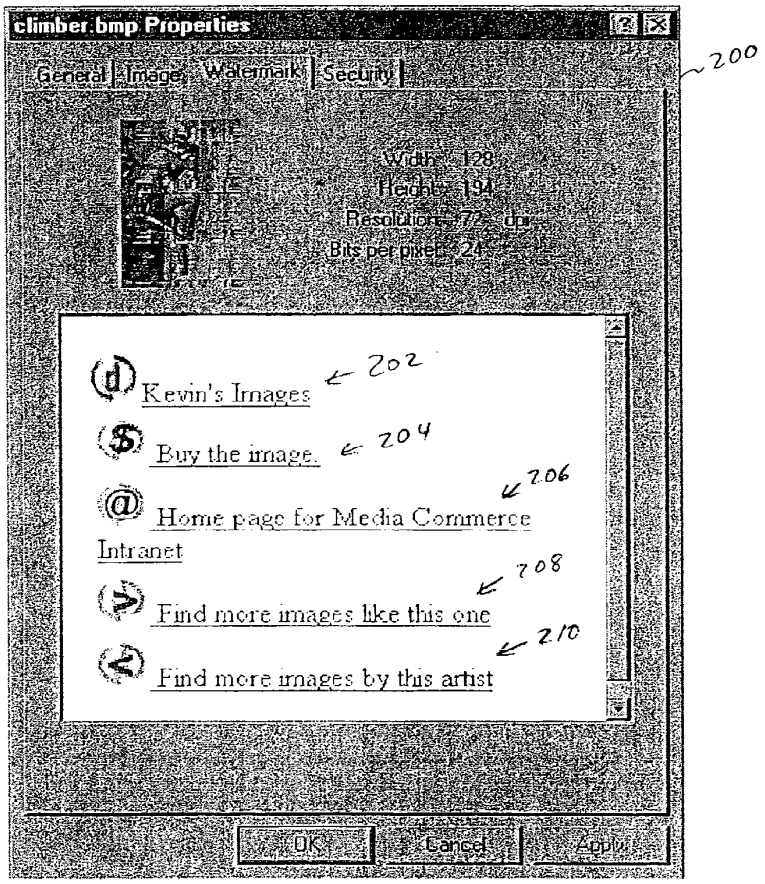

climber.bmp Properties
General
Image
Watermark
Security
Width: 128
Height: 194
Resolution: 72 dpi
Bits per pixel: 24
Kevin's Images
Buy the image.
Home page for Media Commerce Intranet
Find more images like this one
Find more images by this artist
OK
Cancel
Apply
200
202
204
206
208
210

Fig. 2

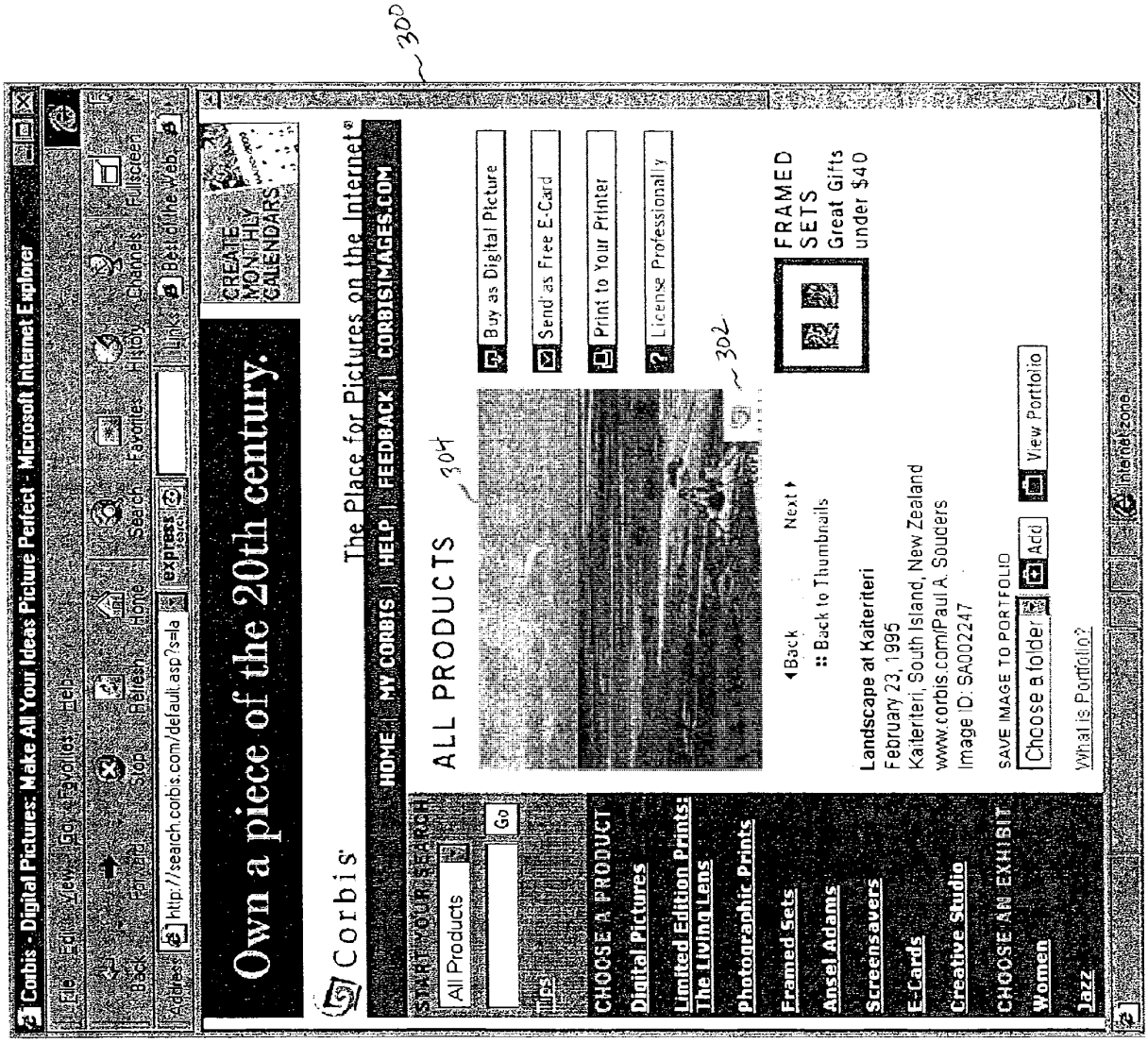

300
Corbis - Digital Pictures: Make All Your Ideas Picture Perfect - Microsoft Internet Explorer
Own a piece of the 20th century.
Corbis
The Place for Pictures on the Internet
HOME | MY CORBIS | HELP | FEEDBACK | CORBISIMAGES.COM
CREATE MONTHLY CALENDARS
ALL PRODUCTS
304
302
Buy as Digital Picture
Send as Free E-Card
Print to Your Printer
License Professionally
Back
Next
Back to Thumbnails
FRAMED SETS
Great Gifts under $40
Landscape at Kaiteriteri
February 23, 1995
Kaiteriteri, South Island, New Zealand
www.corbis.com/Paul A. Souders
Image ID: SA002247
SAVE IMAGE TO PORTFOLIO
Choose a folder
Add
View Portfolio
What is Portfolio?
CHOOSE A PRODUCT
Digital Pictures
Limited Edition Prints:
The Living Lens
Photographic Prints
Framed Sets
Ansel Adams
Screensavers
E-Cards
Creative Studio
CHOOSE AN EXHIBIT
Women
Jazz
All Products
Go
Tips

Fig. 3

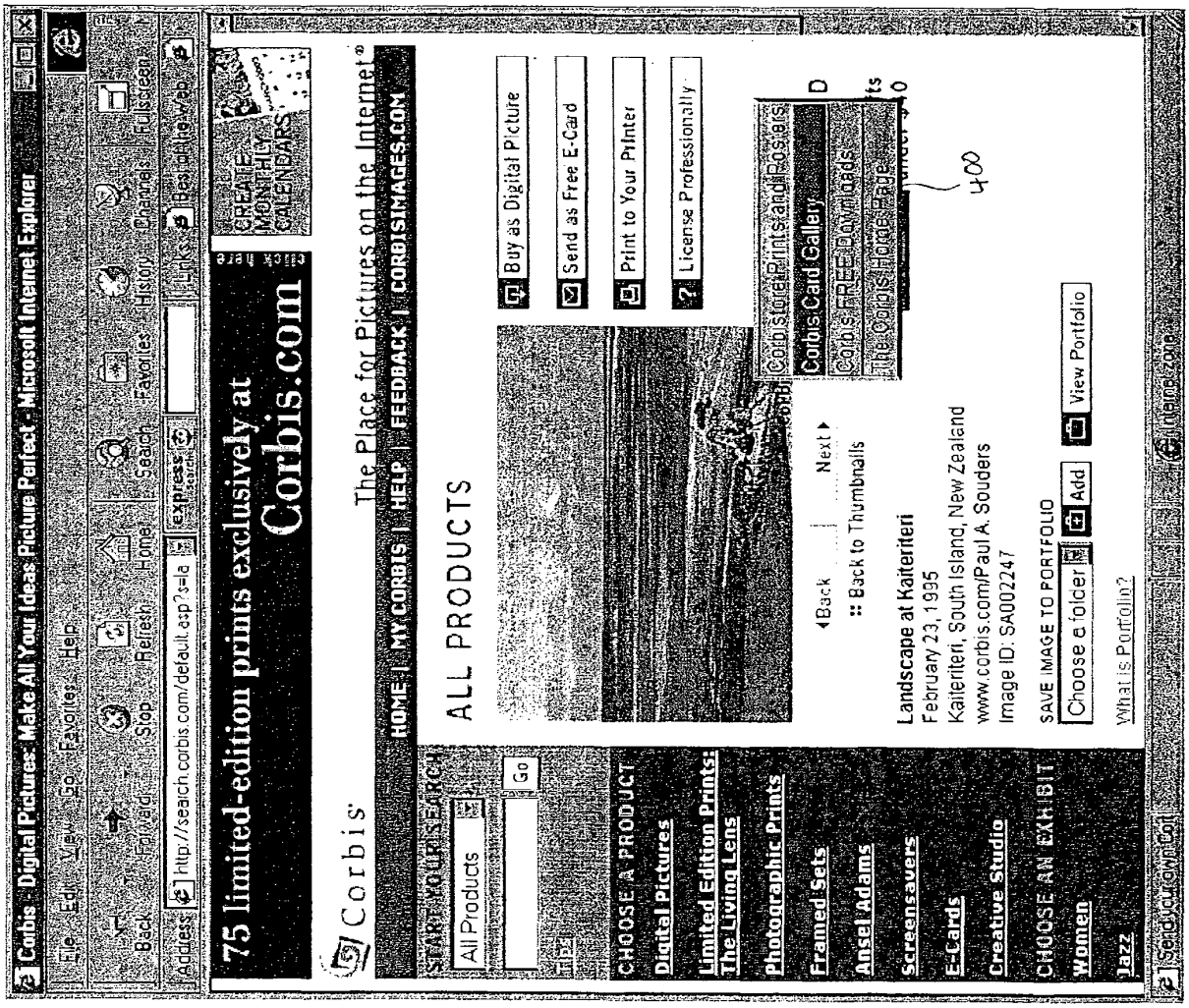

Corbis - Digital Pictures: Make All Your Ideas Picture Perfect - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen
Address http://search.corbis.com/default.asp?s=le
Links Best of the Web
75 limited-edition prints exclusively at Corbis.com
Corbis
The Place for Pictures on the Internet
HOME | MY CORBIS | HELP | FEEDBACK | CORBISIMAGES.COM
CREATE MONTHLY CALENDARS
START YOUR SEARCH
All Products
Go
CHOOSE A PRODUCT
Digital Pictures
Limited Edition Prints
The Living Lens
Photographic Prints
Framed Sets
Ansel Adams
Screensavers
E-Cards
Creative Studio
CHOOSE AN EXHIBIT
Women
Jazz
ALL PRODUCTS
Buy as Digital Picture
Send as Free E-Card
Print to Your Printer
License Professionally
Corbis Store: Prints and Posters
Corbis Card Gallery
Corbis FREE Downloads
The Corbis Home Page
Back
Next
Back to Thumbnails
Landscape at Kaiteriteri
February 23, 1995
Kaiteriteri, South Island, New Zealand
www.corbis.com/Paul A. Souders
Image ID: SA002247
SAVE IMAGE TO PORTFOLIO
Choose a folder
Add
View Portfolio
What is Portfolio?
Done
Internet zone

Fig. 4

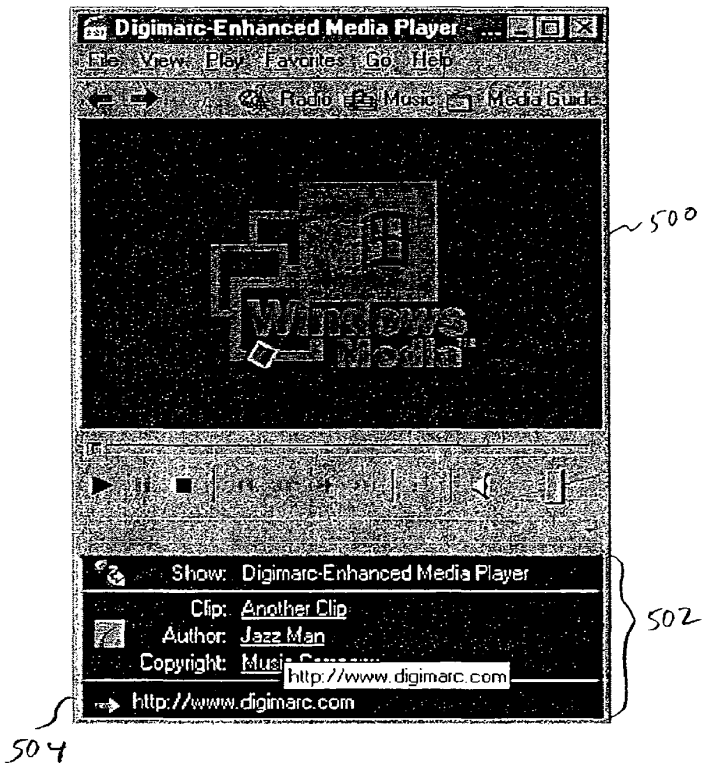

Digimarc-Enhanced Media Player
File View Play Favorites Go Help
Radio Music Media Guide
Windows Media
500
Show: Digimarc-Enhanced Media Player
Clip: Another Clip
Author: Jazz Man
Copyright:
http://www.digimarc.com
http://www.digimarc.com
502
504

Fig. 5

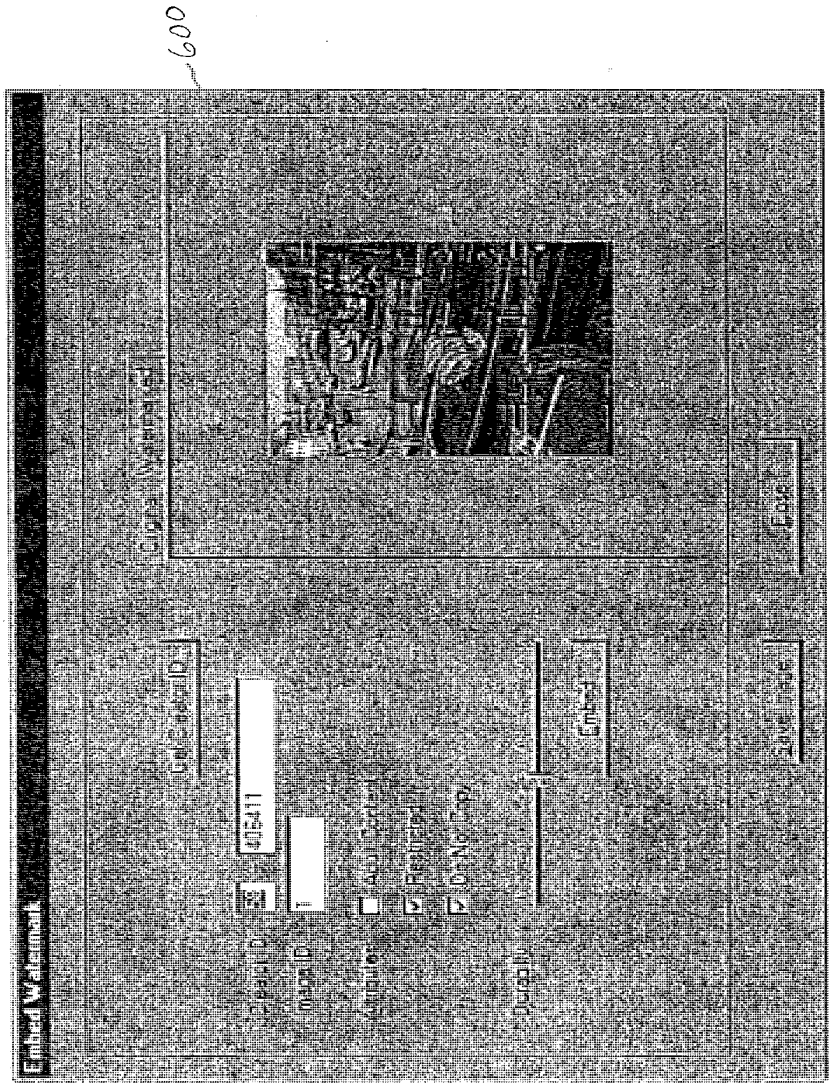


Fig. 6

FIG. 7

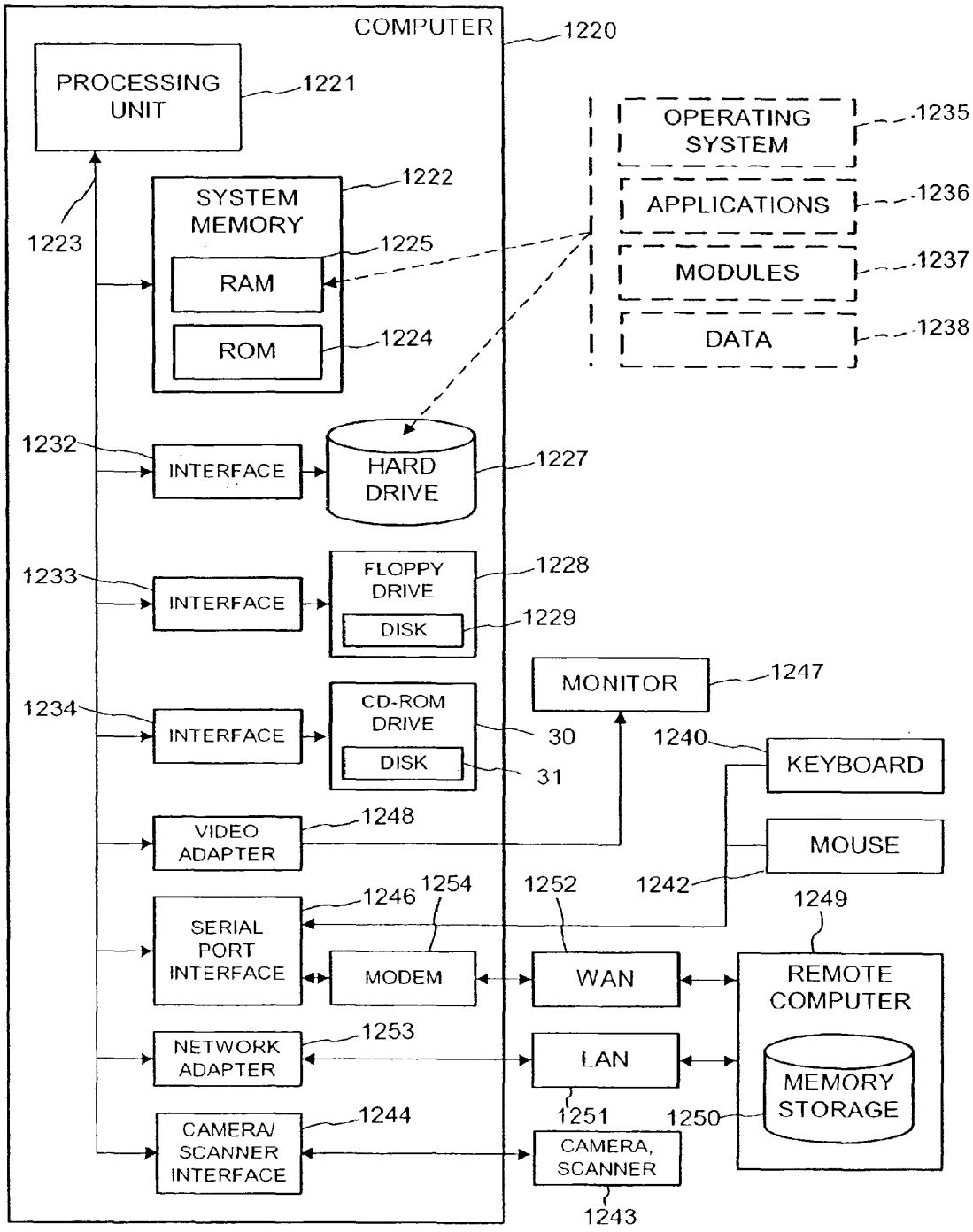
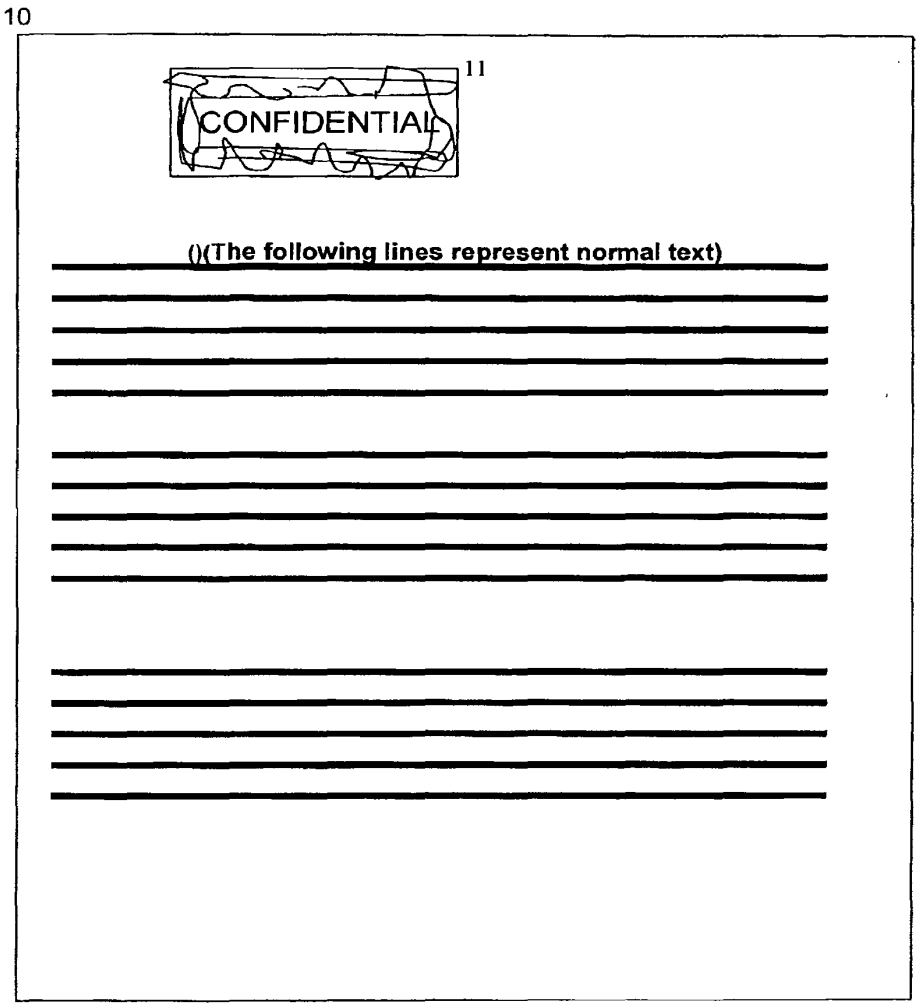

COMPUTER
1220
PROCESSING UNIT
1221
SYSTEM MEMORY
1222
1223
RAM
1225
ROM
1224
OPERATING SYSTEM
1235
APPLICATIONS
1236
MODULES
1237
DATA
1238
1232
INTERFACE
HARD DRIVE
1227
1233
INTERFACE
FLOPPY DRIVE
1228
DISK
1229
1234
INTERFACE
CD-ROM DRIVE
30
DISK
31
MONITOR
1247
1240
KEYBOARD
MOUSE
1242
VIDEO ADAPTER
1248
SERIAL PORT INTERFACE
1246
1254
1252
1249
MODEM
WAN
REMOTE COMPUTER
NETWORK ADAPTER
1253
LAN
MEMORY STORAGE
1251
1250
CAMERA/ SCANNER INTERFACE
1244
CAMERA, SCANNER
1243

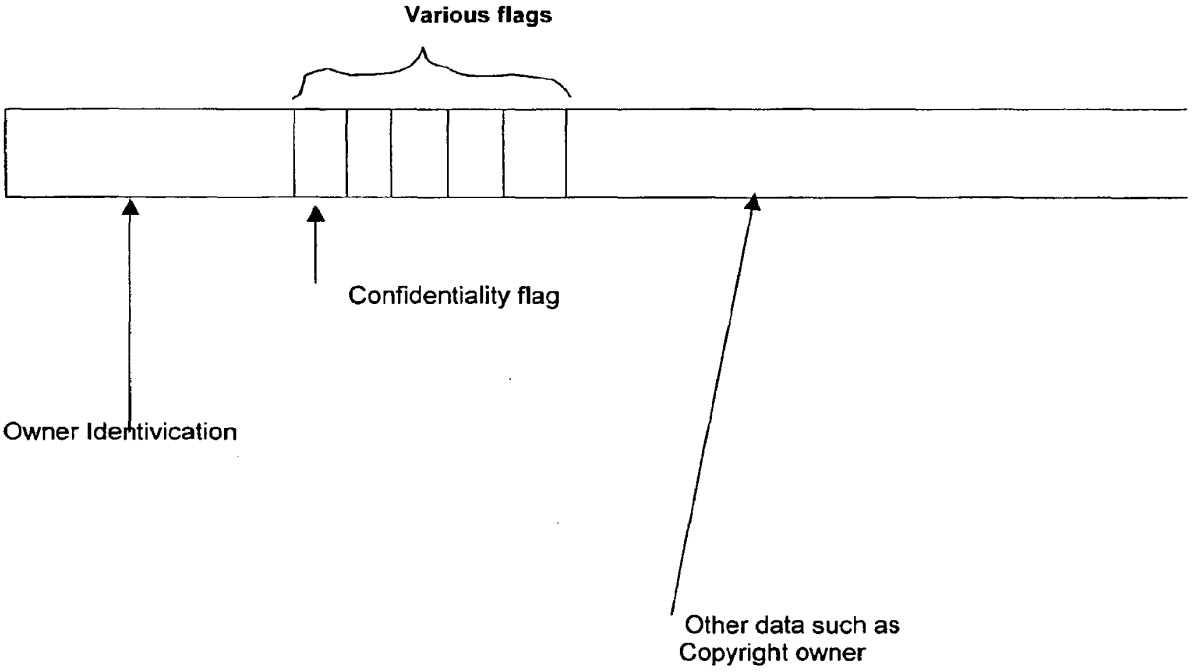
Figure 1 — 8



EWG-076 Figure 1 2-7-00.doc
02/19/00 0

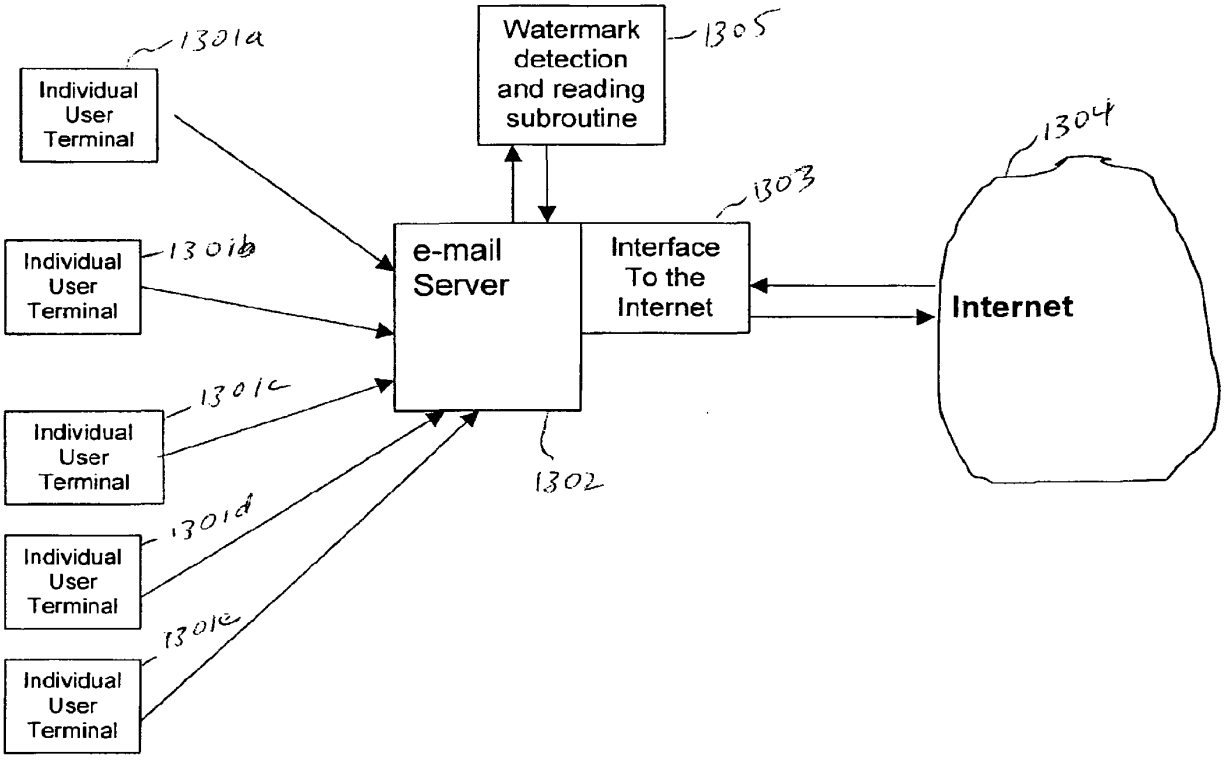
Figure 2 of 9

Fields in a typical watermark payload



EWG-076 Figure 2 2-7-00.doc
02/19/00 11

Figure 3



EWG-076 Figure 3 2-19-00.doc 02/19/00

Figure 4 (Re-11)

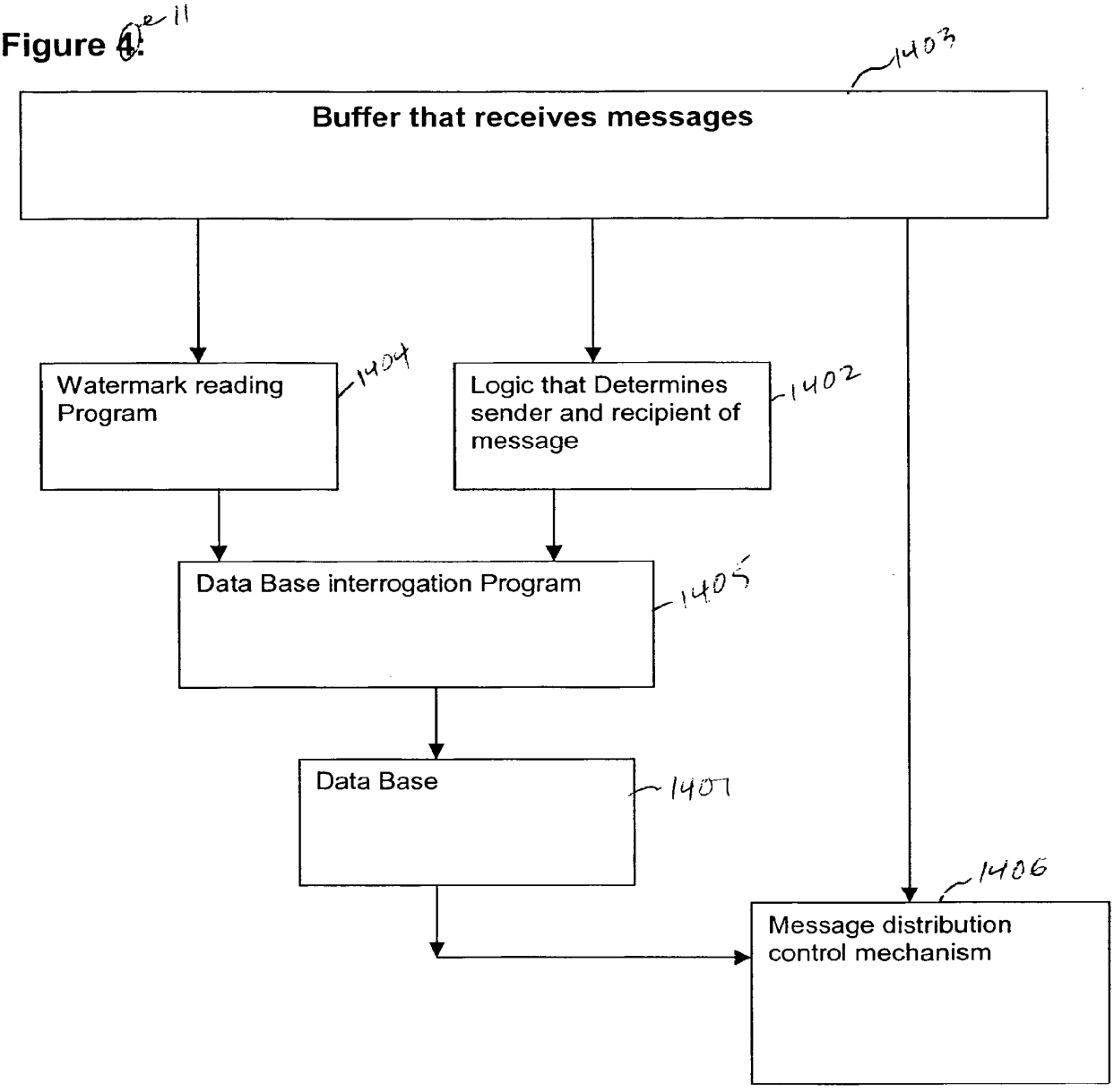
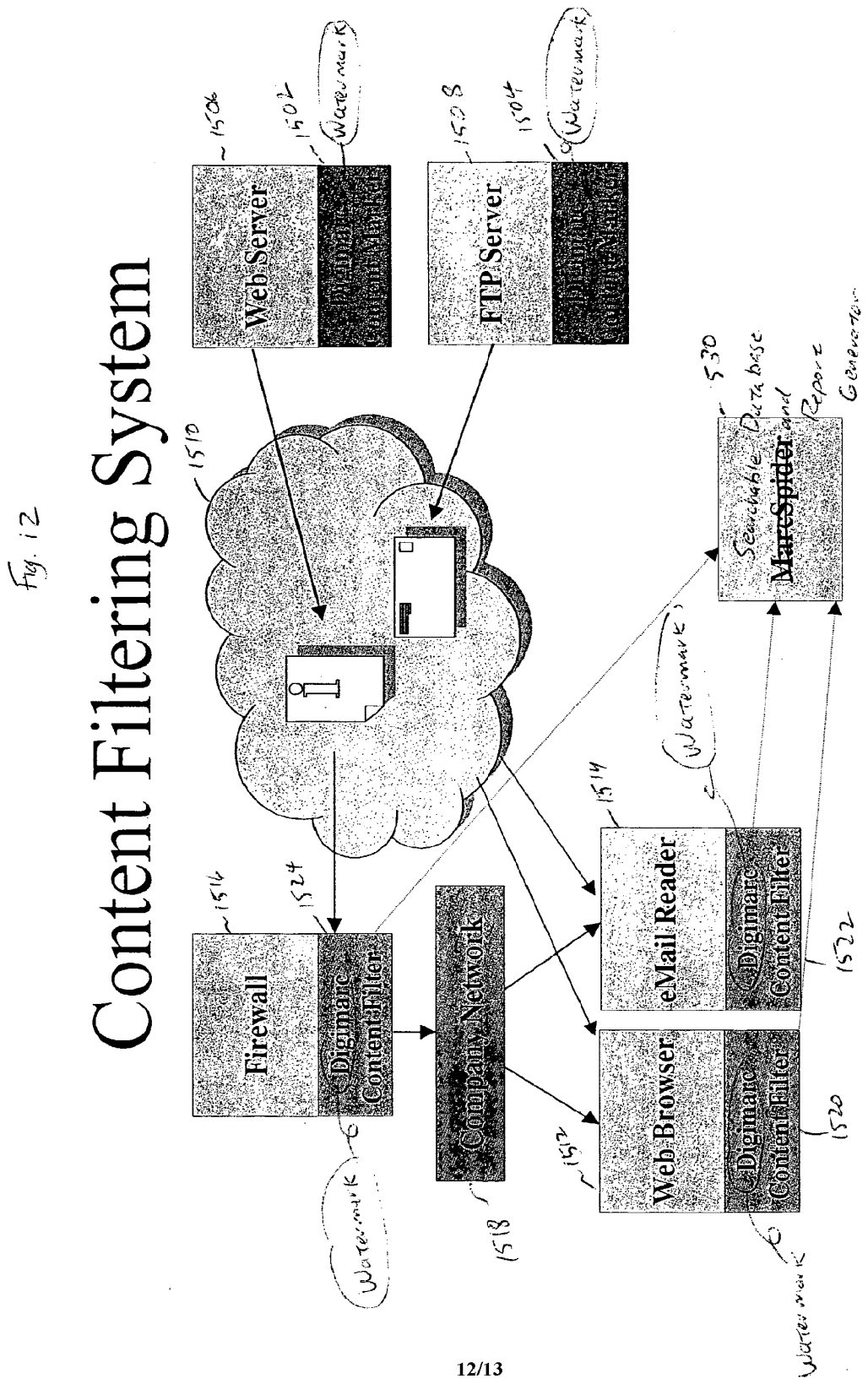


Fig. 12


Content Filtering System
Web Server
FTP Server
Digimarc Content Filter
Firewall
Company Network
eMail Reader
Web Browser
Searchable Data base and Report Generator
MarcSpider
Watermark

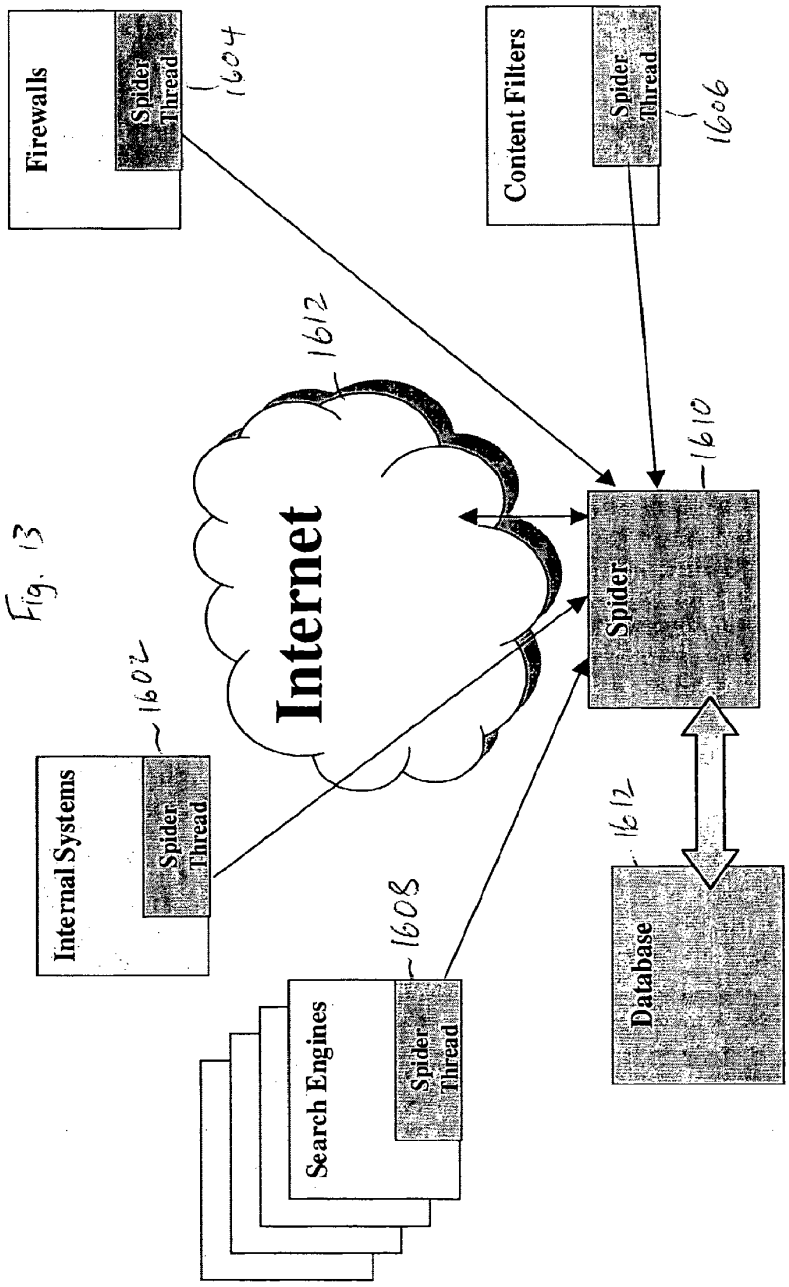

Firewalls
Spider Thread
1604
Content Filters
Spider Thread
1606
1612
Internet
1610
Spider
Internal Systems
Spider Thread
1602
1608
Search Engines
Spider Thread
1612
Database

Fig. 13

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/04812

A. CLASSIFICATION OF SUBJECT MATTER
IPC(7) : G06F 13/00, 15/16; H04L 9/00
US CL : 709/217; 380/28; 345/335
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 709/217; 380/28, 54, 279; 345/335, 339; 283/72, 113; 713/161, 176

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,956,716 A (KENNER et al) 21 SEPTEMBER 1999, column 20, lines 22-column 27, lines 61, column 32, lines 38-column 33, lines 12, column 23, lines 25-49, column 32, lines 64-column 33, lines 12, column 23, lines 66-column 24, lines 14, column 27, lines 64, lines 64-column 28, lines 17, column 29, lines 27-35	1-20
Y	US 5,841,978 A (RHOADS) 24 NOVEMBER 1998, column 2, lines 9-26, column 58, lines 34-column 59, lines 47	1-20

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
01 June 2001 (01.06.2001)	28 JUN 2001
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer Tadesse Hailu Telephone No. (703) 306-2799

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

tional application No.
PCT/US01/04812

Box I Observations where certain claims were found unsearchable (Continuation of Item 1 of first sheet)

This international report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claim Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claim Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claim Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of Item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:
GROUP 1: CLAIMS 1-20; GROUP II: CLAIMS 21-25; GROUP III: CLAIMS 26-32

The inventions listed as Groups I, II and III do not relate to a single inventive concept under PCT Rule 13.1 because, under PCT Rule 13.2, they lack the same or corresponding special technical features for the following reasons: the special technical feature of Group I invention is the browser system claimed therein while the special technical feature of the Group II invention is the electronic massaging system and the feature of Group III invention is content filtering system. Since the special technical feature of the Group I invention is not present in Groups II and III inventions being claimed and the special technical feature of the Group II invention is not present in Groups I and III inventions being claimed and the special technical feature of the Group III invention is not present in Group I and II inventions being claimed, unity of invention is lacking.

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.: 1-20

Remark on Protest
☐ The additional search fees were accompanied by the applicant's protest.
☐ No protest accompanied the payment of additional search fees.

Form PCT/ISA/210 (continuation of first sheet(1)) (July 1998)

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 October 2001 (04.10.2001)

PCT

(10) International Publication Number
WO 01/73586 A2

(51) International Patent Classification[7]: G06F 17/00

(21) International Application Number: PCT/US01/10151

(22) International Filing Date: 29 March 2001 (29.03.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/537,530 29 March 2000 (29.03.2000) US

(71) Applicant: **DIGITAL:CONVERGENCE CORPORATION** [US/US]; Suite 600, 9101 N. Central Expressway, Dallas, TX 75231 (US).

(72) Inventor: **PHILYAW, Jeffrey, Jovan**; 5968 West Northwest Highway, No. 1813, Dallas, TX 75225 (US).

(74) Agents: **HOWISON, Gregory, M.** et al.; Howison, Chauza, Handley & Arnott, L.L.P., P.O. Box 741715, Dallas, TX 75374-1715 (US).

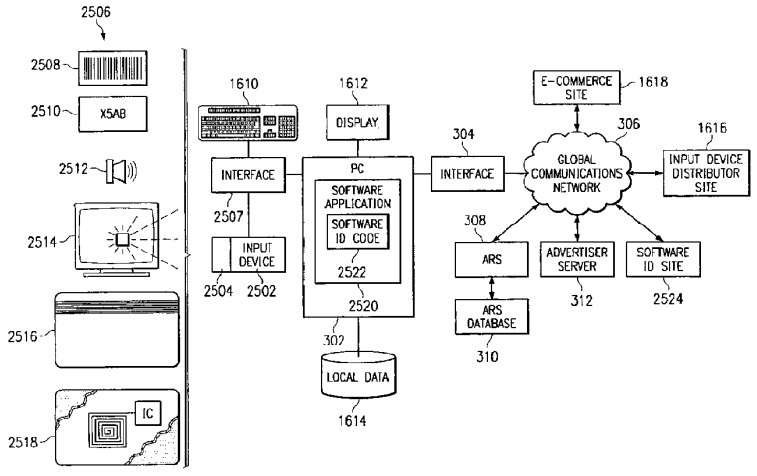
(81) Designated States *(national)*: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD AND APPARATUS FOR ACCESSING A REMOTE LOCATION BY SENSING A MACHINE-RESOLVABLE CODE



(57) Abstract: A method for controlling a computer wherein one or more remote locations disposed on a network are accessed in response to sensing a machine-resolvable code. A computer disposed on a network is operably connected to an input device for sensing a machine-resolvable code. A software application which includes a software identification code runs on the computer. In response to sensing a machine-resolvable code with the input device, the computer accesses at least one remote location corresponding to the software identification code. The one or more remote locations accessed may then return remote information to the computer for presentation.

WO 01/73586 A2

METHOD AND APPARATUS FOR ACCESSING A REMOTE LOCATION BY SENSING A MACHINE-RESOLVABLE CODE

TECHNICAL FIELD OF THE INVENTION

This invention is related to a method of computer control. In one aspect, it relates to a system for automatically directing a web browser application on a computer at a first location to retrieve and display information from a remote location in response to sensing a machine-resolvable code present at the first location.

CROSS REFERENCE TO RELATED APPLICATION

This application is a Continuation-In-Part of pending U.S. Patent Application No. 09/378,221, filed August 19, 1999 and entitled "METHOD AND APPARATUS FOR ACCESSING A REMOTE LOCATION BY SCANNING AN OPTICAL CODE" (Atty Dkt No. PHLY-24,669), which is a Continuation-In-Part of pending U.S. Patent Application Serial No. 09/151,530 and entitled, "METHOD FOR CONTROLLING COMPUTERS THROUGH A RADIO/TELEVISION COMMUNICATION HUB" (Atty Dkt No. PHLY-24,398) filed on September 11, 1998.

BACKGROUND OF THE INVENTION

With the growing numbers of computer users connecting to the "Internet," many companies are seeking the substantial commercial opportunities presented by such a large user base. For example, one technology which exists allows a television ("TV") signal to trigger a computer response in which the consumer will be guided to a personalized web page. The source of the triggering signal may be a TV, video tape recorder, or radio. For example, if a viewer is watching a TV program in which an advertiser offers viewer voting, the advertiser may transmit a unique signal within the television signal which controls a program known as a "browser" on the viewer's computer to automatically display the advertiser's web page. The viewer then simply makes a selection which is then transmitted back to the advertiser.

In order to provide the viewer with the capability of responding to a wide variety of companies using this technology, a database of company information and Uniform Resource Locator ("URL") codes is necessarily maintained in the viewer's computer, requiring continuous updates. URLs are short strings of data that identify resources on the Internet: documents, images, downloadable files, services, electronic mailboxes, and other resources. URLs make resources available under a variety of naming schemes and access methods such as HTTP, FTP, and Internet mail, addressable in the same simple way. URLs reduce the tedium of "login to this server, then issue this magic command ..." down to a single click. The Internet uses URLs to specify the location of files on other servers. A URL includes the type of resource being accessed (e.g., Web, gopher, FTP), the address of the server, and the location of the file. The URL can point to any file on any networked computer. Current technology requires the viewer to perform periodic updates to obtain the most current URL database. This aspect of the current technology is cumbersome since the update process requires downloading information to the viewer's computer. Moreover, the likelihood for error in performing the update, and the necessity of redoing the update in the event of a later computer crash, further complicates the process. Additionally, current technologies are limited in the number of companies which may be stored in the database. This is a significant limitation since world-wide access presented by the Internet and the increasing

number of companies connecting to perform on-line E-commerce necessitates a large database.

SUMMARY OF THE INVENTION

The present invention disclosed and claimed herein comprises, in one aspect thereof, a method for controlling a computer wherein one or more remote locations disposed on a network are accessed in response to sensing a machine-resolvable code. A computer disposed on a network is operably connected to an input device for sensing a machine-resolvable code. A software application which includes a software identification code runs on the computer. In response to sensing a machine-resolvable code with the input device, the computer accesses at least one remote location corresponding to the software identification code. The one or more remote locations accessed may then return remote information to the computer for presentation.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention and the advantages thereof, reference is now made to the following description taken in conjunction with the accompanying Drawings in which:

FIGURE 1 illustrates a block diagram of the preferred embodiment;

FIGURE 2 illustrates the computer components employed in this embodiment;

FIGURE 3 illustrates system interactions over a global network;

FIGURES 4a-4e illustrate the various message packets transmitted between the source PC and network servers used in the preferred embodiment;

FIGURE 5 is a flowchart depicting operation of the system according to the preferred embodiment;

FIGURE 6 illustrates a flowchart of actions taken by the Advertiser Reference Server ("ARS") server;

FIGURE 7 illustrates a flowchart of the interactive process between the source computer and ARS;

FIGURE 8 illustrates a web browser page receiving the modified URL/advertiser product data according to the preferred embodiment;

FIGURE 9 illustrates a simplified block diagram of the disclosed embodiment;

FIGURE 10 illustrates a more detailed, simplified block diagram of the embodiment of FIGURE 9;

FIGURE 11 illustrates a diagrammatic view of a method for performing the routing operation;

FIGURE 12 illustrates a block diagram of an alternate embodiment utilizing an optical region in the video image for generating the routing information;

FIGURE 13 illustrates a block diagram illustrating the generation of a profile with the disclosed embodiment;

FIGURE 14 illustrates a flowchart for generating the profile and storing at the ARS;

FIGURE 15 illustrates a flowchart for processing the profile information when information is routed to a user;

FIGURE 16 illustrates a general block diagram of a disclosed embodiment;

FIGURE 17 illustrates the conversion circuit of the wedge interface;

FIGURE 18 illustrates a sample message packet transmitted from the user PC to the ARS;

FIGURE 19 illustrates a more detailed block diagram of the routing of the message packets between the various nodes;

FIGURE 20 illustrates a block diagram of a browser window, according to a disclosed embodiment;

FIGURE 21 illustrates a diagrammatic view of information contained in the ARS database;

FIGURE 22 illustrates a flowchart of the process of receiving information from the user's perspective;

FIGURE 23 illustrates a flowchart according to the ARS;

FIGURE 24 illustrates a flowchart of the process performed at the E-commerce node;

FIGURE 25 illustrates a general block diagram of another disclosed embodiment;

FIGURE 26 illustrates a sample message packet transmitted from the first PC to the ARS;

FIGURE 27 illustrates another sample message packet transmitted from the first PC to the ARS;

FIGURE 28 illustrates a more detailed block diagram of the routing of the message packets between the various network locations;

FIGURE 29 illustrates a block diagram of a browser window, according to another disclosed embodiment;

FIGURE 30 illustrates a diagrammatic view of information contained in the ARS database according to another disclosed embodiment;

FIGURE 31 illustrates a flowchart of the process of receiving information from the user's perspective;

FIGURE 32 illustrates a flowchart of the process according to the ARS; and

FIGURE 33 illustrates a flowchart of the process performed at the software-specific network location.

DETAILED DESCRIPTION OF THE INVENTION

Referring now to FIGURE 1, there is illustrated a block diagram of a system for controlling a personal computer ("PC") 112 via an audio tone transmitted over a wireless system utilizing a TV. In the embodiment illustrated in FIGURE 1, there is provided a transmission station 101 and a receive station 117 that are connected via a communication link 108. The transmission station 101 is comprised of a television program source 104, which is operable to generate a program in the form of a broadcast signal comprised of video and audio. This is transmitted via conventional techniques along channels in the appropriate frequencies. The program source is input to a mixing device 106, which mixing device is operable to mix in an audio signal. This audio signal is derived from an audio source 100 which comprises a coded audio signal which is then modulated onto a carrier which is combined with the television program source 104. This signal combining can be done at the audio level, or it can even be done at the RF level in the form of a different carrier. However, the preferred method is to merely sum the audio signal from the modulator 102 into the audio channel of the program that is generated by the television program source 104. The output thereof is provided from the mixing device 106 in the form of broadcast signal to an antenna 107, which transmits the information over the communication link 108 to an antenna 109 on the receive side.

On the receive side of the system, a conventional receiver 110, such as a television is provided. This television provides a speaker output which provides the user with an audible signal. This is typically associated with the program. However, the receiver 110 in the disclosed embodiment, also provides an audio output jack, this being the type RCA jack. This jack is utilized to provide an audio output signal on a line 113 which is represented by an audio signal 111. This line 113 provides all of the audio that is received over the communication link 108 to the PC 112 in the audio input port on the PC 112. However, it should be understood that, although a direct connection is illustrated from the receiver 110 to the PC 112, there actually could be a microphone pickup at the PC 112 which could pick the audio signal up. In the disclosed embodiment, the audio signal generated by the

advertiser data input device 100 is audible to the human ear and, therefore, can be heard by the user. Therefore, no special filters are needed to provide this audio to the PC 112.

The PC 112 is operable to run programs thereon which typically are stored in a program file area 116. These programs can be any type of programs such as word processing programs, application programs, etc. In the disclosed embodiment, the program that is utilized in the system is what is referred to as a "browser." The PC 112 runs a browser program to facilitate the access of information on the network, for example, a global communication network known as the "Internet" or the World-Wide-Web ("Web"). The browser is a hypertext-linked application used for accessing information. Hypertext is a term used to describe a particular organization of information within a data processing system, and its presentation to a user. It exploits the computer's ability to link together information from a wide variety of sources to provide the user with the ability to explore a particular topic. The traditional style of presentation used in books employs an organization of the information which is imposed upon it by limitations of the medium, namely fixed sized, sequential paper pages. Hypertext systems, however, use a large number of units of text or other types of data such as image information, graphical information, video information, or sound information, which can vary in size. A collection of such units of information is termed a hypertext document, or where the hypertext documents employ information other than text, hypermedia documents. Multimedia communications may use the Hypertext Transfer Protocol ("HTTP"), and files or formatted data may use the Hypertext Markup Language ("HTML"). This formatting language provides for a mingling of text, graphics, sound, video, and hypertext links by "tagging" a text document using HTML. Data encoded using HTML is often referred to as an "HTML document," an "HTML page," or a "home page." These documents and other Internet resources may be accessed across the network by means of a network addressing scheme which uses a locator referred to as a Uniform Resource Locator ("URL"), for example, "http://www.digital.com."

The Internet is one of the most utilized networks for interconnecting distributed computer systems and allows users of these computer systems to exchange data all over the world. Connected to the Internet are many private networks, for example, corporate or

commercial networks. Standard protocols, such as the Transport Control Protocol ("TCP") and the Internet Protocol ("IP") provide a convenient method for communicating across these diverse networks. These protocols dictate how data are formatted and communicated. As a characteristic of the Internet, the protocols are layered in an IP stack. At higher levels of the IP stack, such as the application layer (where HTTP is employed), the user information is more readily visible, while at lower levels, such as the network level (where TCP/IP are used), the data can merely be observed as packets or a stream of rapidly moving digital signals. Superimposed on the Internet is a standard protocol interface for accessing Web resources, such as servers, files, Web pages, mail messages, and the like. One way that Web resources can be accessed is by browsers made by *Netscape*® and *Microsoft Internet Explorer*®.

Referring again now to FIGURE 1, the user can load this program with the appropriate keystrokes such that a browser window will be displayed on a display 118. In one embodiment, the user can run the browser program on the PC 112 such that the browser window is displayed on the display 118. While watching a preferred program, the user can also view display 118. When an audio signal is received by the receiver 110 and the encoded information is contained therein that was input thereto by the advertiser, the PC 112 will then perform a number of operations. The first operation, according to the disclosed embodiment, is to extract the audio information within the received audio signal in the form of digital data, and then transmit this digital data to a defined location on the global communication network via a modem connection 114. This connection will be described hereinbelow. This information will be relayed to a proprietary location and the instructions sent back to the PC 112 as to the location of the advertiser associated with the code, and the PC 112 will then effect a communication link to that location such that the user can view on the display 118 information that the advertiser, by the fact of putting the tone onto the broadcast channel, desires the viewer to view. This information can be in the form of interactive programs, data files, etc. In one example, when an advertisement appears on the television, the tone can be generated and then additional data displayed on the display 118. Additionally, a streaming video program could be played on the PC received over the network, which streaming video program is actually longer than the

advertising segment on the broadcast. Another example would be a sports game that would broadcast the tone in order to allow a user access to information that is not available over the broadcast network, such as additional statistics associated with the sports program, etc.

By utilizing the system described herein with respect to the disclosed embodiment of FIGURE 1, an advertiser is allowed the ability to control a user's PC 112 through the use of tones embedded within a program audio signal. As will described hereinbelow, the disclosed embodiment utilizes particular routing information stored in the PC 112 which allows the encoded information in the received audio signal to route this information to a desired location on the network, and then allow other routing information to be returned to the PC 112 for control thereof to route the PC 112 to the appropriate location associated with that code.

Referring now to FIGURE 2, there is illustrated a computer 204, similar to computer 112, connected to display information on display 118. The computer 204 comprises an internal audio or "sound" card 206 for receiving the transmitted audio signal through receive antenna 109 and receiver 110. The sound card 206 typically contains analog-to-digital circuitry for converting the analog audio signal into a digital signal. The digital signal may then be more easily manipulated by software programs. The receiver 110 separates the audio signal from the video signal. A special trigger signal located within the transmitted advertiser audio signal triggers proprietary software running on the computer 204 which launches a communication application, in this particular embodiment, the web browser application located on the PC 204. Coded advertiser information contained within the audio signal is then extracted and appended with the address of a proprietary server located on the communication network.

The remote server address is in the form of a URL. This appended data, in addition to other control codes, is inserted directly into the web browser application for automatic routing to the communication network. The web browser running on PC 204, and communicating to the network with an internal modem 208, in this embodiment, transmits the advertiser information to the remote server. The remote server cross-references the

advertiser product information to the address of the advertiser server located on the network. The address of the advertiser server is routed back through the PC 204 web browser to the advertiser server. The advertiser product information is returned to PC 204 to be presented to the viewer on display 118. In this particular embodiment, the particular advertiser product information displayed is contained within the advertiser's web page 212. As mentioned above, the audio signal is audible to the human ear. Therefore the audio signal, as emitted from the TV speakers, may be input to the sound card 206 via a microphone. Furthermore, the audio signal need not be a real-time broadcast, but may be on video tapes, CDs, DVD, or other media which may be displayed at a later date. With the imminent implementation of high definition digital television, the audio signal output from the TV may also be digital. Therefore, direct input into a sound card for A/D purposes may not be necessary, but alternative interfacing techniques to accommodate digital-to-digital signal formats would apply.

Referring now to FIGURE 3, there is illustrated a source PC 302, similar to PCs 204 and 112, connected to a global communication network ("GCN") 306 through an interface 304. In this embodiment, the audio signal 111 is received by PC 302 through its sound card 206. The audio signal 111 comprises a trigger signal which triggers proprietary software into launching a web browser application residing on the PC 302. The audio signal 111 also comprises advertiser product information which is extracted and appended with URL information of an Advertiser Reference Server ("ARS") 308. The ARS 308 is a system disposed on the GCN 306 that is defined as the location to which data in the audio signal 111 is to be routed. As such, data in the audio signal 111 will always be routed to the ARS 308, since a URL is unique on the GCN 306. Connected to the ARS 308 is a database 310 of product codes and associated manufacturer URLs. The database 310 undergoes a continual update process which is transparent to the user. As companies sign-on, i.e., subscribe, to this technology, manufacturer and product information is added to the database 310 without interrupting operation of the source PC 302 with frequent updates. When the advertiser server address URL is obtained from the ARS database 310, it and the request for the particular advertiser product information are automatically routed back through the web browser on PC 302, over to the respective advertiser server for retrieval of the advertiser

product information to the PC 302. Additionally, although the disclosed invention discusses a global communication network, the system is also applicable to LANs, WANs, and peer-to-peer network configurations. It should be noted that the disclosed architecture is not limited to a single source PC 302, but may comprise a plurality of source PCs, e.g., PC 300 and PC 303. Moreover, a plurality of ARS 308 systems and advertiser servers 312 may be implemented, e.g., ARS 314, and advertiser server A 316, respectively.

The information transactions, in general, which occur between the networked systems of this embodiment, over the communication network, are the following. The web browser running on source PC 302 transmits a message packet to the ARS 308 over Path "A." The ARS 308 decodes the message packet and performs a cross-reference function with product information extracted from the received message packet to obtain the address of an advertiser server 312. A new message packet is assembled comprising the advertiser server 312 address, and sent back to the source PC 302 over Path "B." A "handoff" operation is performed whereby the source PC 302 browser simply reroutes the information on to the advertiser server 312 over Path "C," with the appropriate source and destination address appended. The advertiser server 312 receives and decodes the message packet. The request-for-advertiser-product-information is extracted and the advertiser 312 retrieves the requested information from its database for transmission back to the source PC 302 over Path "D." The source PC 302 then processes the information, i.e., for display to the viewer. The optional Path "E" is discussed hereinbelow. It should be noted that the disclosed methods are not limited to only browser communication applications, but may accommodate, with sufficient modifications by one skilled in the art, other communication applications used to transmit information over the Internet or communication network.

Referring now to FIGURE 4a, the message packet 400 sent from the source PC 302 to ARS 308 via Path "A" comprises several fields. One field comprises the URL of the ARS 308 which indicates where the message packet is to be sent. Another field comprises the advertiser product code or other information derived from the audio signal 111, and any additional overhead information required for a given transaction. The product code provides a link to the address of the advertiser server 312, located in the database 310. Yet

another field comprises the network address of the source PC 302. In general, network transmissions are effected in packets of information, each packet providing a destination address, a source address, and data. These packets vary depending upon the network transmission protocol utilized for communication. Although the protocols utilized in the disclosed embodiment are of a conventional protocol suite commonly known as TCP/IP, it should be understood that any protocols providing the similar basic functions can be used, with the primary requirement that a browser can forward the routing information to the desired URL in response to keystrokes being input to a PC. Within the context of this disclosure, "message packet" shall refer to and comprise the destination URL, product information, and source address, even though more than a single packet must be transmitted to effect such a transmission.

Upon receipt of the message packet 400 from source PC 302, ARS 308 processes the information in accordance with instructions embedded in the overhead information. The ARS 308 specifically will extract the product code information from the received packet 400 and, once extracted, will then decode this product code information. Once decoded, this information is then compared with data contained within the ARS advertiser database 310 to determine if there is a "hit." If there is no "hit" indicating a match, then information is returned to the browser indicating such. If there is a "hit," a packet 402 is assembled which comprises the address of the source PC 302, and information instructing the source PC 302 as to how to access, directly in a "handoff" operation, another location on the network, that of an advertiser server 312. This type of construction is relatively conventional with browsers such as *Netscape*® and *Microsoft Internet Explorer*® and, rather than displaying information from the ARS 308, the source PC 302 can then access the advertiser server 312. The ARS 308 transmits the packet 402 back to source PC 302 over Path "B." Referring now to FIGURE 4b, the message packet 402 comprises the address of the source PC 302, the URL of the advertiser server 312 embedded within instructional code, and the URL of the ARS 308.

Upon receipt of the message packet 402 by the source PC 302, the message packet 402 is disassembled to obtain pertinent routing information for assembly of a new message

packet 404. The web browser running on source PC 302 is now directed to obtain, over Path "C," the product information relevant to the particular advertiser server 312 location information embedded in message packet 404. Referring now to FIGURE 4c, the message packet 404 for this transaction comprises the URL of the advertiser server 312, the request-for-product-information data, and the address of the source PC 302.

Upon receipt of the message packet 404 from source PC 302, advertiser server 312 disassembles the message packet 404 to obtain the request-for-product-information data. The advertiser server 312 then retrieves the particular product information from its database, and transmits it over Path "D" back to the source PC 302. Referring now to FIGURE 4d, the message packet 406 for this particular transaction comprises the address of the source PC 302, the requested information, and the URL of the advertiser server 312.

Optionally, the ARS 308 may make a direct request for product information over Path "E" to advertiser server 312. In this mode, the ARS 308 sends information to the advertiser server 312 instructing it to contact the source PC 302. This, however, is unconventional and requires more complex software control. The message packet 408 for this transaction is illustrated in FIGURE 4e, which comprises the URL of the advertiser server 312, the request-for-product-information data, and the address of the source PC 302. Since product information is not being returned to the ARS 308, but directly to the source PC 302, the message packet 408 requires the return address to be that of the source PC 302. The product information is then passed directly to PC 302 over Path "D."

Referring now to FIGURE 5, the method for detecting and obtaining product information is as follows. In decision block 500, a proprietary application running resident on a source computer PC 302 (similar to PC 204) monitors the audio input for a special trigger signal. Upon detection of the trigger signal, data following the trigger signal is decoded for further processing, in function block 502. In function block 504, the data is buffered for further manipulation. In decision block 506, a determination is made as to whether the data can be properly authenticated. If not, program flow continues through the "N" signal to function block 520 where the data is discarded. In function block 522, the

program then signals for a retransmission of the data. The system then waits for the next trigger signal, in decision block 500. If properly authenticated in decision block 506, program flow continues through the "Y" signal path where the data is then used to launch the web browser application, as indicated in function block 508. In function block 510, the web browser receives the URL data, which is then automatically routed through the computer modem 208 to the network interface 304 and ultimately to the network 306. In function block 514, the ARS 308 responds by returning the URL of advertiser server 312 to the PC 302. In function block 516, the web browser running on the source PC 302, receives the advertiser URL information from the ARS 308, and transmits the URL for the product file to the advertiser server 312. In block 518, the advertiser server 312 responds by sending the product information to the source PC 302 for processing.

The user may obtain the benefits of this architecture by simply downloading the proprietary software over the network. Other methods for obtaining the software are well-known; for example, by CD, diskette, or pre-loaded hard drives.

Referring now to FIGURE 6, there is illustrated a flowchart of the process the ARS 308 may undergo when receiving the message packet 400 from the source PC 302. In decision block 600, the ARS 308 checks for the receipt of the message packet 400. If a message packet 400 is not received, program flow moves along the "N" path to continue waiting for the message. If the message packet 400 is received, program flow continues along path "Y" for message processing. Upon receipt of the message packet 400, in function block 602, the ARS 308 decodes the message packet 400. The product code is then extracted independently in function block 604 in preparation for matching the product code with the appropriate advertiser server address located in the database 310. In function block 606, the product code is then used with a lookup table to retrieve the advertiser server 312 URL of the respective product information contained in the audio signal data. In function block 608, the ARS 308 then assembles message packet 402 for transmission back to the source PC 302. Function block 610 indicates the process of sending the message packet 402 back to the source PC 302 over Path "B."

Referring now to FIGURE 7, there is illustrated a flowchart of the interactive processes between the source PC 302 and the advertiser server 312. In function block 700, the source PC 302 receives the message packet 402 back from the ARS 308 and begins to decode the packet 402. In function block 702, the URL of the advertiser product information is extracted from the message packet 402 and saved for insertion into the message packet 404 to the advertiser server 312. The message packet 404 is then assembled and sent by the source PC 302 over Path "C" to the advertiser server 312, in function block 704. While the source PC 302 waits, in function block 706, the advertiser server 312 receives the message packet 404 from the source PC 302, in function block 708, and disassembles it. The product information location is then extracted from the message packet 404 in function block 710. The particular product information is retrieved from the advertiser server 312 database for transmission back to the source PC 302. In function block 712, the product information is assembled into message packet 406 and then transmitted back to the source PC 302 over Path "D." Returning to the source PC 302 in function block 714, the advertiser product information contained in the message packet 406 received from the advertiser server 312, is then extracted and processed in function block 716.

Referring now to FIGURE 8, after receipt of a trigger signal, a web browser application on a source PC 302 is automatically launched and computer display 800 presents a browser page 802. Proprietary software running on the source PC 302 processes the audio signal data after being digitized through the sound card 206. The software appropriately prepares the data for insertion directly into the web browser by extracting the product information code and appending keystroke data to this information. First, a URL page 804 is opened in response to a Ctrl-O command added by the proprietary software as the first character string. Opening URL page 804 automatically positions the cursor in a field 806 where additional keystroke data following the Ctrl-O command will be inserted. After URL page 804 is opened, the hypertext protocol preamble http:// is inserted into the field 806. Next, URL information associated with the location of the ARS 308 is inserted into field 806. Following the ARS 308 URL data are the characters /? to allow entry of variables immediately following the /? characters. In this embodiment, the variable following is the

product information code received in the audio signal. The product code information also provides the cross-reference information for obtaining the advertiser URL from the ARS database 310. Next, a carriage return is added to send the URL/product data and close the window 804. After the message packet 400 is transmitted to the ARS 308 from the source PC 302, transactions from the ARS308, to the source PC 302, to the advertiser server 312, and back to the source PC 302, occur quickly and are transparent to the viewer. At this point, the next information the viewer sees is the product information which was received from the advertiser server 312.

Referring now to FIGURE 9, there is illustrated a block diagram of a more simplified embodiment. In this embodiment, a video source 902 is provided which is operable to provide an audio output on an audio cable 901 which provides routing information referred to by reference numeral 904. The routing information 904 is basically information contained within the audio signal. This is an encoded or embedded signal. The important aspect of the routing information 904 is that it is automatically output in realtime as a function of the broadcast of the video program received over the video source 902. Therefore, whenever the program is being broadcast in realtime to the user 908, the routing information 904 will be output whenever the producer of the video desires it to be produced. It should be understood that the box 902 representing the video source could be any type of media that will result in the routing information being output. This could be a cassette player, a DVD player, an audio cassette, a CD ROM or any such media. It is only important that this is a program that the producer develops which the user 908 watches in a continuous or a streaming manner. Embedded within that program, at a desired point selected by the producer, the routing information 904 is output.

The audio information is then routed to a PC 906, which is similar to the PC 112 in FIGURE 1. A user 908 is interfaced with the PC to receive information thereof, the PC 906 having associated therewith a display (not shown). The PC 906 is interfaced with a network 910, similar to the network 306 in FIGURE 3. This network 910 has multiple nodes thereon, one of which is the PC 906, and another of which is represented by a network node 912 which represents remote information. The object of the present embodiment is to

access remote information for display to the user 908 by the act of transmitting from the video program in block 902 the routing information 904. This routing information 904 is utilized to allow the PC 906 which has a network "browser" running thereon to "fetch" the remote information at the node 912 over the network 910 for display to the user 908. This routing information 904 is in the form of an embedded code within the audio signal, as was described hereinabove.

Referring now to FIGURE 10, there is illustrated a more detailed block diagram of the embodiment of FIGURE 9. In this embodiment, the PC 906 is split up into a couple of nodes, a first PC 1002 and a second PC 1004. The PC 1002 resides at the node associated with the user 908, and the PC 1004 resides at another node. The PC 1004 represents the ARS 308 of FIGURE 3. The PC 1004 has a database 1006 associated therewith, which is basically the advertiser database 310. Therefore, there are three nodes on the network 910 necessary to implement the disclosed embodiment, the PC 1002, the PC 1004 and the remote information node 912. The routing information 904 is utilized by the PC 1002 for routing to the PC 1004 to determine the location of the remote information node 912 on the network 910. This is returned to the PC 1002 and a connection made directly with the remote information node 912 and the information retrieved therefrom to the user 908. The routing information 904 basically constitutes primary routing information.

Referring now to FIGURE 11, there is illustrated a diagrammatic view of how the network packet is formed for sending the primary routing information to the PC 1004. In general, the primary routing information occupies a single field which primary routing information is then assembled into a data packet with the secondary routing information for transfer to the network 910. This is described hereinabove in detail.

Referring now to FIGURE 12, there is illustrated an alternate embodiment to that of FIGURE 9. In this embodiment, the video source 902 has associated therewith an optical region 1202, which optical region 1202 has disposed therein an embedded video code. This embedded video code could be relatively complex or as simple as a grid of dark and white regions, each region in the grid able to have a dark color for a logic "1" or a white region for

a logic "0." This will allow a digital value to be disposed within the optical region 1202. A sensor 1204 can then be provided for sensing this video code. In the example above, this would merely require an array of optical detectors, one for each region in the grid to determine whether this is a logic "1" or a logic "0" state. One of the sensed video is then output to the PC 906 for processing thereof to determine the information contained therein, which information contained therein constitutes the primary routing information 904. Thereafter, it is processed as described hereinabove with reference to FIGURE 9.

Referring now to FIGURE 13, there is illustrated a block diagram for an embodiment wherein a user's profile can be forwarded to the original subscriber or manufacturer. The PC 906 has associated therewith a profile database 1302, which profile database 1302 is operable to store a profile of the user 908. This profile is created when the program, after initial installation, requests profile information to be input in order to activate the program. In addition to the profile, there is also a unique ID that is provided to the user 908 in association with the browser program that runs on the PC 906. This is stored in a storage location represented by a block 1304. This ID 1304 is accessible by a remote location as a "cookie" which is information that is stored in the PC 906 in an accessible location, which accessible location is actually accessible by the remote program running on a remote node.

The ARS 308, which basically constitutes the PC 1004 of FIGURE 10, is operable to have associated therewith a profile database 1308, which profile database 1308 is operable to store profiles for all of the users. The profile database 1308 is a combination of the stored in profile database 1302 for all of the PCs 906 that are attachable to the system. This is to be distinguished from information stored in the database 310 of the ARS 308, the advertiser's database, which contains intermediate destination tables. When the routing information in the primary routing information 904 is forwarded to the ARS 308 and extracted from the original data packet, the lookup procedure described hereinabove can then be performed to determine where this information is to be routed. The profile database 1302 is then utilized for each transaction, wherein each transaction in the form of the

routing information received from the primary routing information 904 is compared to the destination tables of database 310 to determine what manufacturer is associated therewith.

The associated ID 1304 that is transmitted along with the routing information in primary routing information 904 is then compared with the profile database 1308 to determine if a profile associated therewith is available. This information is stored in a transaction database 1310 such that, at a later time, for each routing code received in the form of the information in primary routing information 904, there will associated therewith the IDs 1304 of each of the PCs 906. The associated profiles in database 1308, which are stored in association with IDs 1304, can then be assembled and transmitted to a subscriber as referenced by a subscriber node 1312 on the network 910. The ARS 308 can do this in two modes, a realtime mode or a non-realtime mode. In a realtime mode, each time a PC 906 accesses the advertiser database 310, that user's profile information is uploaded to the subscriber node 1312. At the same time, billing information is generated for that subscriber 1312 which is stored in a billing database 1316. Therefore, the ARS 308 has the ability to inform the subscriber 1312 of each transaction, bill for those transactions, and also provide to the subscriber 1312 profile information regarding who is accessing the particular product advertisement having associated therewith the routing information field 904 for a particular routing code as described hereinabove. This information, once assembled, can then be transmitted to the subscriber 1312 and also be reflected in billing information and stored in the billing information database 1316.

Referring now to FIGURE 14, there is illustrated a flowchart depicting the operation for storing the profile for the user. The program is initiated in a block 1402 and then proceeds to a function block 1404, wherein the system will prompt for the profile upon initiation of the system. This initiation is a function that is set to activate whenever the user initially loads the software that he or she is provided. The purpose for this is to create, in addition to the setup information, a user profile. Once the user is prompted for this, then the program will flow to a decision block 1406 to determine whether the user provides basic or detailed information. This is selectable by the user. If selecting basic, the program will flow to a function block 1408 wherein the user will enter basic information such as name

and serial number and possibly an address. However, to provide some incentive to the user to enter more information, the original prompt in function block 1404 would have offers for such things as coupons, discounts, etc., if the user will enter additional information. If the user selects this option, the program flows from the decision block 1406 to a function block 1410. In the function block 1410, the user is prompted to enter specific information such as job, income level, general family history, demographic information and more. There can be any amount of information collected in this particular function block.

Once all of the information is collected, in either the basic mode or the more specific mode, the program will then flow to a function block 1412 where this information is stored locally. The program then flows to a decision block 1414 to then go on-line to the host or the ARS 308. In general, the user is prompted to determine whether he or she wants to send this information to the host at the present time or to send it later. If he or she selects the "later" option, the program will flow to a function block 1415 to prompt the user at a later time to send the information. In the disclosed embodiment, the user will not be able to utilize the software until the profile information is sent to the host. Therefore, the user may have to activate this at a later time in order to connect with the host.

If the user has selected the option to upload the profile information to the host, the program will flow to the function block 1416 to initiate the connect process and then to a decision block 1418 to determine if the connection has been made. If not, the program will flow along a "N" path to a time to decision block 1420 which will timeout to an error block 1422 or back to the input of the connect decision block 1418. The program, once connected, will then flow along a "Y" path from decision block 1418 to a function block 1428 to send the profile information with the ID of the computer or user to the host. The ID is basically, as described hereinabove, a "cookie" in the computer which is accessed by the program when transmitting to the host. The program will then flow to a function block 1430 to activate the program such that it, at later time, can operate without requiring all of the setup information. In general, all of the operation of this flowchart is performed with a "wizard" which steps the user through the setup process. Once complete, the program will flow to a Done block 1432.

Referring now to FIGURE 15, there is illustrated a flowchart depicting the operation of the host when receiving a transaction. The program is initiated at a Start block 1502 and then proceeds to decision block 1504, wherein it is determined whether the system has received a routing request, i.e., the routing information 904 in the form of a tone, etc., embedded in the audio signal, as described hereinabove with respect to FIGURE 9. The program will loop back around to the input of decision block 1504 until the routing request has been received. At this time, the program will flow along the "Y" path to a function block 1506 to receive the primary routing information and the user ID. Essentially, this primary routing information is extracted from the audio tone, in addition to the user ID. The program then flows to a function block 1508 to look up the manufacturer URL that corresponds to the received primary routing information and then return the necessary command information to the originating PC 108 in order to allow that PC 108 to connect to the destination associated with the primary routing information. Thereafter, the program will flow to a function block 1510 to update the transaction database 1310 for the current transaction. In general, the routing information 904 will be stored as a single field with the associated IDs. The profile database 1308, as described hereinabove, has associated therewith detailed profiles of each user on the system that has activated their software in association with their ID. Since the ID was sent in association with the routing information, what is stored in the transaction database 1310 is the routing code, in association with all of the IDs transmitted to the system in association with that particular routing code. Once this transaction database 1310 has been updated, as described hereinabove, the transactions can be transferred back to the subscriber at node 312 with the detailed profile information from the profile database 1308.

The profile information can be transmitted back to the subscriber or manufacturer at the node 312 in realtime or non-realtime. A decision block 1512 is provided for this, which determines if the delivery is realtime. If realtime, the program will flow along a "Y" path to a function block 1514 wherein the information will be immediately forwarded to the manufacturer or subscriber. The program will then flow to a function block 1516 wherein the billing for that particular manufacturer or subscriber will be updated in the billing database 1316. The program will then flow into an End block 1518. If it was non-realtime,

the program moves along the "N" path to a function block 1520 wherein it is set for a later delivery and it is accrued in the transaction database 1310. In any event, the transaction database 1310 will accrue all information associated with a particular routing code.

With a realtime transaction, it is possible for a manufacturer to place an advertisement in a magazine or to place a product on a shelf at a particular time. The manufacturer can thereafter monitor the times when either the advertisements are or the products are purchased. Of course, they must be scanned into a computer which will provide some delay. However, the manufacturer can gain a very current view of how a product is moving. For example, if a cola manufacturer were to provide a promotional advertisement on, for example, television, indicating that a new cola was going to be placed on the shelf and that the first 1000 purchasers, for example, scanning their code into the network would receive some benefit, such as a chance to win a trip to some famous resort in Florida or some other incentive, the manufacturer would have a very good idea as to how well the advertisement was received. Further, the advertiser would know where the receptive markets were. If this advertiser, for example, had placed the television advertisement in ten cities and received overwhelming response from one city, but very poor response from another city, he would then have some inclination to believe that either the one poor-response city was not a good market or that the advertising medium he had chosen was very poor. Since the advertiser can obtain a relatively instant response and also content with that response as to the demographics of the responder, very important information can be obtained in a relatively short time.

It should be noted that the disclosed embodiment is not limited to a single source PC 302, but may encompass a large number of source computers connected over a global communication network. Additionally, the embodiment is not limited to a single ARS 308 or a single advertiser server 312, but may include a plurality of ARS and advertiser systems, indicated by the addition of ARS 314 and advertiser server A 316, respectively. It should also be noted that this embodiment is not limited only to global communication networks, but also may be used with LAN, WAN, and peer-to-peer configurations.

It should also be noted that the disclosed embodiment is not limited to a personal computer, but is also applicable to, for example, a Network Computer ("NetPC"), a scaled-down version of the PC, or any system which accommodates user interaction and interfaces to information resources.

One typical application of the above noted technique is for providing a triggering event during a program, such as a sport event. In a first example, this may be generated by an advertiser. One could imagine that, due to the cost of advertisements in a high profile sports program, there is a desire to utilize this time wisely. If, for example, an advertiser contracted for 15 seconds worth of advertising time, they could insert within their program a tone containing the routing information. This routing information can then be output to the user's PC 302 which will cause the user's PC 302 to, via the network, obtain information from a remote location typically controlled by the advertiser. This could be in the form of an advertisement of a length longer than that contracted for. Further, this could be an interactive type of advertisement. An important aspect to the type of interaction between the actual broadcast program with the embedded routing information and the manufacturer's site is the fact that there is provided information as to the user's PC 302 and a profile of the user themselves. Therefore, an advertiser can actually gain realtime information as to the number of individuals that are watching their particular advertisement and also information as to the background of those individuals, profile information, etc. This can be a very valuable asset to an advertiser.

In another example, the producer of the program, whether it be an on-air program, a program embedded in a video tape, CD-ROM, DVD, or a cassette, can allow the user to automatically access additional information that is not displayed on the screen. For example, in a sporting event, various statistics can be provided to the user from a remote location, merely by the viewer watching the program. When these statistics are provided, the advertiser can be provided with profile information and background information regarding the user. This can be important when, for example, the user may record a sports program. If the manufacturer sees that this program routing code is being output from some device at a time later than the actual broadcast itself, this allows the advertisers to actually

see that their program is still being used and also what type of individual is using it. Alternatively, the broadcaster could determine the same and actually bill the advertiser an additional sum for a later broadcast. This is all due to the fact that the routing information automatically, through a PC and a network, will provide an indication to the advertiser the time at which the actual information was broadcast.

The different type of medium that can be utilized with the above embodiment are such things as advertisements, which are discussed hereinabove, contests, games, news programs, education, coupon promotional programs, demonstration media (demos), and photographs, all of which can be broadcast on a private site or a public site. This all will provide the ability to allow realtime interface with the network and the remote location for obtaining the routed information and also allow for realtime billing and accounting.

Referring now to FIGURE 16, there is illustrated a general block diagram of a disclosed embodiment. A bar code scanning input device 1600 is provided by a input device distributor to customers and is associated with that distributor via a input device ID stored therein. The input device 1600 is either sold or freely distributed to customers for use with their personal computing systems. Since more and more products are being sold using bar codes, it can be appreciated that a user having the input device 1600 can scan bar codes of a multitude of products in order to obtain more information. Information about these products can be made immediately available to the user from the manufacturer for presentation by the user's computer 302. Beyond simply displaying information about the product in which the user is interested, the input device distributor may include additional advertising information for display to the user such as information about other promotions or products provided or sold by the input device distributor. Similarly, advertisers may provide catalogs of advertisements or information in newspapers or periodicals where the user simply scans the bar code associated with the advertisement using the input device 1600 to obtain further information. There is provided a paper source 1602 having contained thereon an advertisement 1604 and an associated bar code 1606. (Note that the disclosed concept is not limited to scanning of bar codes 1606 from paper sources 1602, but is also

operable to scan a bar code 1606 on the product itself. Also, the input device 1600 can be any type of device that will scan any type of image having information encoded therein.)

After obtaining the input device 1600 from the input device distributor, the user connects the input device 1600 to their PC 302. During a scanning operation, input device 1600 reads bar code data 1606 and the input device ID into a "wedge" interface 1608 for conversion into keyboard data, which keyboard data is passed therefrom into the keyboard input port of PC 302. The importance of the input device ID will be discussed in more detail hereinbelow.

The wedge interface 1608 is simply an interface box containing circuitry that accommodates inputs from both the scanning input device 1600 and a computer keyboard 1610. This merely allows the information scanned by the input device 1600 to be input into the PC 302. In the disclosed embodiment, the wedge interface 1608 will convert any information. The data output from the input device 1600 is passed into the wedge interface 1608 for conversion into keyboard data which is readily recognizable by the PC 302. Therefore, the input device 1600 is not required to be connected to a separate port on the PC 302. This data is recognized as a sequence of keystrokes. However, the output of the input device 1600 can be input in any manner compatible with the PC 302. When not receiving scanner data, the wedge interface 1608 simply acts as a pass-through device for keyboard data from the keyboard 1610. In any case, the information is ultimately processed by a processor in the PC 302 and can be presented to the user on a display 1612. The wedge interface 1608 is operable to provide a decoding function for the bar code 1606 and conversion thereof to keystroke input data.

In operation, the product code of a product is provided in the form of a bar code 1606. This bar code 1606 is the "link" to a product. The disclosed embodiment is operable to connect that product information contained in the bar code 1606 with a web page of the manufacturer of that product by utilizing the bar code 1606 as the product "identifier." The program operating on the PC 302 provides routing information to the ARS 308 after launching the browser on the PC 302 and connecting to the ARS 308 over the GCN 306,

which ARS 308 then performs the necessary steps to cause the browser to connect to the manufacturer web site, while also providing for an accounting step, as will be described in more detail hereinbelow.

The bar code 1606 by itself is incompatible with any kind of network for the purposes of communication therewith. It is primarily provided for a retail-type setting. Therefore, the information contained in the bar code 1606, by itself, does not allow for anything other than identification of a product, assuming that one has a database 1614 containing information as to a correlation between the product and the bar code 1606.

The wedge interface 1608 is operable to decode the bar code 1606 to extract the encoded information therein, and append to that decoded bar code information relating to an ID for the input device 1600. This information is then forwarded to the ARS 308 by the resident program in the PC 302. This is facilitated by intermediate routing information stored in the program indicating to which node on the GCN 306 the scanned bar code information is to be sent, i.e., to the ARS 308. It is important to note that the information in the bar code 1606 must be converted from its optical image to numerical values which are then ultimately input to the keyboard input port of PC 302 and converted into data compatible with communication software residing on the PC 302 (in this case, HTML language for insertion into a browser program). When the scanned information is input to the PC 302, the resident program launches the browser program and then assembles a communication packet comprised of the URL of the ARS 308, the input device ID and the user ID. If another type of communications program were utilized, then it would have to be converted into language compatible with that program. Of course, a user could actually key in the information on the bar code 102 and then append the appropriate intermediate routing information thereafter. As will be described hereinbelow, the intermediate routing information appended thereto is the URL of the ARS 308 disposed on the GCN 306.

As part of the configuration for using the input device 1600, the PC 302 hosts input device software which is operable to interpret data transmitted from the input device 1600, and to create a message packet having the scanned product information and input device ID,

routing information, and a user ID which identifies the user location of the input device 1600. The input device software loads at boot-up of the PC 302 and runs in the background. In response to receiving a scanned bar code 1606, the wedge interface 1608 outputs a keystroke code (e.g., ALT-F10) to bring the input device program into the foreground for interaction by the operating system. The input device program then inserts the necessary information into the browser program. The message packet is then transmitted to interface 304 across the global communication network 306 to the ARS 308. The ARS 308 interrogates the message packet and performs a lookup function using the ARS database 310. If a match is found between particular parameters of the message packet, a return message packet is sent back to the PC 302 for processing.

The input device program running on PC 302 functions to partition the browser window displayed to the user into several individual areas. This is for the purpose of preparing to present to the user selected information in each of the individual areas (also called "framing"). The selected information comprises the product information which the user requested by scanning the bar code 1606 using the input device 1600, information about the input device distributor which establishes the identity of the company associated with that particular input device 1600, and at least one or more other frames which may be advertisements related to other products that the input device distributor sells. Note that the advertisements displayed by the input device distributor may be related to the product of interest or totally unrelated. For example, if a user scans the bar code 1606 of a soda from Company A, the input device distributor may generate an advertisement of a new soft drink being marketed by Company A, that it sells. On the other hand, the input device distributor may also structure the display of information to the user such that a user requesting product information of a Product X may get the requested information of Product X along with advertisements for a competing item Product Y. Essentially, the input device distributor is free to generate any advertisement to the user in response to the user requesting product information.

The return message packet transmitted from the ARS 308 to the PC 302 is then transmitted back across the GCN 306 to the advertiser server 312. The advertiser server 312

restructures the message packet and appends the particular product information for transmission back to the PC 302. Upon receiving the particular advertiser information from advertiser server 312, the PC 302 then retransmits a message to the input device distributor site 1616 and E-commerce site 1618 to obtain the information that needs to be framed in the browser window displayed to the user.

Therefore, the input device 1600 is associated with the input device distributor by way of a input device ID such that scanning a product bar code 1606 in order to obtain information about that particular product generates one or more responses from one or more remote sites disposed on the GCN 306. Stored in the input device 1600 is the input device ID which establishes its relationship to the input device distributor. Proprietary input device software running on the PC 302 operates to decode scanned bar code information and the input device ID received from the input device 1600 and wedge interface 1608, and also provides a unique user ID for establishing the location of the user of the input device 1600. The input device software also assembles message packets and works in conjunction with the on-board communication software (e.g., a browser) to automatically route the message packets across the GCN 306 such that the one or more remote sites disposed on the GCN 306 return information to be framed for presentation to the user.

Referring now to FIGURE 17, there is illustrated a conversion circuit of the wedge interface. A microcontroller 1700 provides conversion of the data from the input device 1600 and controls interfacing of the keyboard 1610 and input device 1600 with the PC 302. The microcontroller 1700 has contained therein a memory 1702 or it can have external memory. There are provided a plurality of input device interfaces 1704 to the input device 1600, a plurality of PC interfaces 1706 to the PC 302, and plurality of keyboard interfaces 1708 to the keyboard 1610. In general, the input device interfaces 1704 comprise a serial data line, a ground line, and a power line. Similarly, the keyboard interfaces 1708 comprise a serial data line, a ground line, a clock line, and a power line. The PC 302 provides a clock line, a power line, a serial data, and a ground line for input to the microcontroller 1700. The microcontroller 1700 is operable to receive signals from the keyboard 1610 and transfer the signals to the PC 302 as keyboard signals. Operation with the keyboard 1610 is essentially a

"pass-through" procedure. Data output from the keyboard 1610 is already in keyboard format, and therefore requires no conversion by the wedge interface 1608. With respect to the input device 1600, the serial data is not compatible with a keyboard 1610 and, therefore, it must be converted into a keyboard format in order to allow input thereof to the keyboard input of the PC 302.

The microcontroller 1700 performs this function after decoding this bar code information, and conversion of this bar code information into an appropriate stream of data which is comprised of the bar code information and the appended URL. This appended URL will be pre-stored in the memory 1702 and is programmable at the time of manufacture. It is noted that the memory 1702 is illustrated as being contained within the microcontroller 1702 to provide a single chip solution. However, this could be external memory that is accessible by the microcontroller 1702. Therefore, the microcontroller 1700 provides an interface between the input device 1600 and the keyboard 1610 to the PC 302 which allows the input device 1600 to receive coded information and convert it to keyboard strokes or, alternatively, to merely pass-through the keystrokes from the keyboard 1610. Therefore, the user need not install any type of plug-in circuit board into the motherboard of the PC 302 in order to provide an interface to the input device 1600; rather, the user need only utilize the already available keyboard port in order to input the appropriate data into the system.

In this particular disclosed embodiment, the microcontroller 1700 comprises a PIC16C73 microcontroller by *Microchip Technologies*™. The PIC16C73 device is a low cost CMOS 8-bit microcontroller with an integrated analog-to-digital converter. The PIC16C73 device, as illustrated in the disclosed embodiment, has 192 bytes of RAM and 4k x 4 of EPROM memory. The microcontroller 1700 can accommodate asynchronous or synchronous inputs from input devices connected to it. In this disclosed embodiment, communication to the keyboard 1610 is synchronous while it is asynchronous when communicating with input device 1600.

It should be noted that, although in this particular embodiment bar code information of the bar code 1606 is input into the keyboard input port of the PC 302, disclosed methods may also be advantageously utilized with high speed port architectures such as Universal Serial Bus ("USB") and IEEE 1394.

Bar codes are structured to be read in either direction. Timing considerations need to be addressed because of the variety of individuals scanning the bar code introduce a wide variety of scan rates. Bar codes use bars of varying widths. The presence of a black bar generates a positive pulse, and the absence of a black bar generates no pulse. Each character of a conventional bar code has associated therewith seven pulses or bars. Depending on the width of the bars, the time between pulses varies. In this disclosed embodiment, the interface circuitry 1608 performs a "running" calculation of the scan time based upon the rising edge of the pulses commencing with the leader or header information. The minimum and maximum scans times are calculated continuously in software with the interface 1608 during the scanning process to ensure a successful scan by the user.

Referring now to FIGURE 18, there is illustrated a sample message packet transmitted from the user's PC 302 to the ARS 308. The message packet 1800 comprises a number of bits of information including the bar code information 1802 obtained from the user scanning the bar code 1606 with the input device 1600; the input device ID 1804 which is embedded in a memory in the input device 1600 and identifies it with a particular input device distributor; and a user ID 1806 which is derived from the software running on the PC 302 and which identifies uniquely with the user location. Note that the message packet includes other necessary information for the proper transmission for point to point.

Referring now to FIGURE 19, there is illustrated a more detailed block diagram of the routing of the message packets in order to present the framed information to the user. As is mentioned hereinabove, when the user scans a bar code 1606 using the input device 1600, a input device program running on the user PC 302 is operable to interpret the information output by the input device 1600 and generate a message packet for transmission over the GCN 306. The input device program assembles the message packet such that it is

directed to the ARS 308 disposed on the GCN 306. The message packet contains several pieces of information including the input device ID 1804 which links it to the input device distributor, the user ID 1806 which identifies the particular user using the input device 1600, and bar code information 1802 describing a particular product of interest to the user. This message from the PC 302 is transmitted over a path 1900 to the ARS 308 where the ARS database 310 is accessed to cross reference the ID information 1804 and bar code information 1802 to a particular advertiser and input device distributor. The ARS 308 returns a message packet over a path 1902 to the user PC 302 which contains routing information as to the location of various other sites disposed on the GCN 306, for example, the advertiser server 312 and input device distributor site 1616.

It can be appreciated that other information can also be provided by the ARS 308 which more closely targets the particular user of the input device 1600. For example, if it is known that a particular input device 1600 is sold in a certain geographic area, this information can be useful in targeting the particular user with certain advertising information relevant to that geographic area. In any case, the information returned from the ARS 308 over path 1902 provides enough information for the input device program running on the user PC 302 to identify a number of other sites disposed on the GCN 306. The user PC 302 then processes the return message packet and routes another message packet over a path 1904 to the advertiser server 312. The advertiser server 312 then returns product information of the particular product in which the user was interested back to the user PC 302 over a path 1906. Similarly, the user PC 302 routes information (e.g., the URL of the input device distributor site and the user profile) to the input device distributor site 1616 over a path 1908 in order to obtain information back over a path 1910 for framing any banners which identify the input device distributor. Additionally, the user PC 302 forwards a message packet to the E-commerce site 1618 over a path 1912 in order to return information regarding any particular advertisements the input device distributor wants to display to the user. The advertisements are returned to the PC 302 over a path 1914.

Referring now to FIGURE 20, there is illustrated a block diagram of a browser window according to the disclosed embodiment. The browser window 2000 is partitioned

into a plurality of areas for framing specific information. A bar code area 2002 displays that product information in which the user was interested; an input device-specific area 2004 displays information about the input device distributor; and an E-commerce area 2006 displays advertising information that the input device distributor selects for display according to this particular user and input device 1600. As mentioned hereinabove, a program operable to process scanned bar code information with the unique input device 1600 develops the browser window by partitioning it into specific areas for the framing of information. Therefore, information returned from the E-commerce site 1608 is passed through the GCN 306 to the particular E-commerce frame 2006. Similarly, information about the particular product of interest is returned from the advertiser site 312 across the GCN 306 to the particular bar code specific area 2002. Information placed in the input device specific area 2004 is information about the input device distributor which is returned from the input device distributor site 1616 across GCN 306.

Referring now to FIGURE 21, there is illustrated a structure of information contained in the ARS database. The ARS database 310 contains a variety of information required to properly interrogate and assemble packets for obtaining information from the various sites disposed on the GCN 306. The ARS database 310 has a database structure 2100 which contains addresses for the web sites containing the product information requested by the user when scanning the bar code 1606 with the input device 1600. Under a PRODUCT heading 2102 are listed the particular bar codes and associated routing information for addressing the respective server location. For example, the ARS server 308 may contain any number of advertisers having unique URL addresses associated therewith. Therefore, the bar code 1606 of a particular product is associated with a unique URL address which routes any request for information of that product to that particular advertiser's site. Also part of the ARS database structure 2000 is a heading of INPUT DEVICE under which is the input device ID 1804 and the distributor associated with that input device ID 1804.

It can be appreciated that there may be a number of distributors using the disclosed architecture such that each distributor has an ID embedded in the input device 1600 which

uniquely identifies that input device with the particular distributor. Therefore, the unique input device ID 1804 needs to be listed with the respective distributors of that input device 1600 in order to process the information that needs to be framed and displayed to that particular user. Another heading under the ARS database structure 2100 is a user heading 2106 which contains profile information associated with that particular user ID 1806. As mentioned hereinabove, the user ID 1806 is obtained via the input device software running on the PC 302 and upon installation or subsequent configuration may request that the user input certain profile information which may be used to target that particular user with products and services which identify with that user profile. The ARS database structure 2100 also contains an E-commerce heading 2108 which contains information related to the bar code 1606 and an advertisement that may be triggered by the request for that information. For example, any bar code 1606 associated with a paper source 1602 can be associated with the specific information in the ARS database 310. A user wishing to obtain information about a specific soft drink may, in fact, trigger an advertising response of a competitor product. Similarly, the user interested in information about that particular soft drink may also trigger information which is relevant to that particular product or a product which may normally be served in conjunction with that soft drink. Furthermore, if the user profile indicates that this individual has significant interest in finance or insurance, the request for information regarding this particular bar coded product may trigger advertisement from an E-commerce server 1618 related to information about finance and insurance. It should be noted that the information described as contained within the ARS database structure 2100 is not limited to what has been described, but may comprise any number of pieces of information used to present desired information to the computer display of the user.

Referring now to FIGURE 22, there is illustrated a flowchart of the process of receiving information from the user's perspective, and according to the disclosed embodiment. The input device software running on the user's PC 302 runs in the background until activated by output from the input device 1600. Therefore, flow moves to a decision block 2200 where if a scanned input does not occur, flow moves out the "N" path and loops back to the input of decision block 2200. On the other hand, if scanned input

information is received, flow moves out the "Y" path to a function block 2202 where the input device software assembles a message packet containing the bar code information, the input device ID 1804 and the ARS 308 URL address. Additionally, the browser is launched in which this information is placed for transmission to the ARS 308. Flow then moves to a function block 2204 where the browser is partitioned into any number of areas in which information is displayed when obtained from the input device distributor site 1616, the E-commerce site 1618, and the advertiser server 312. It should be known that although three frames are shown in the particular window 2000 of this embodiment, the number of frames displayed in the window 2000 is limited only by the available real estate of the window 2000 area itself.

After the input device software partitions the browser window into one or more frames in preparation of receipt of return information, flow moves to a decision block 2206 where the computer waits for information to be returned from the various sites disposed on the GCN 306. If information is not returned, flow moves out the "N" path and simply loops back to the input to continue monitoring for receipt of the information. If information has been received, flow moves out the "Y" path to a function block 2208 where routing information for each frame (or partitioned area of the window 2000) is inserted into one or more packets for transmission to the various sites. The various sites then return the requested information back to the PC 302, as indicated in function block 2210. Flow is then to a function block 2212 where the proprietary software working in conjunction with the hosted browser places the returned information into the respective frames of the window. The user, viewing the display at PC 302, then perceives a variety of information, one of which is the particular product information which he or she requested, in addition to input device distributor information, and possibly other advertisements based upon the user's profile.

Referring now to FIGURE 23, there is illustrated a flowchart of the process according to the ARS. The ARS 308 is operable to decode and process messages received from the GCN 306. Therefore, flow is to a decision block 2300 where, if bar code information is not received, flow is out the "N" path with loop-back to its input. If bar code

information has been received, flow is to a function block 2302 where a matching process occurs to link the bar-coded product information to its respective manufacturer. The ARS database 310 also associates the URL address of the manufacturer's server. When a match is found, the ARS 308 begins to assemble a message packet of information for transmission back to the PC 302, as indicated in function block 2304. The message packet contains the product information and the URL address of the manufacturer's website. Flow then moves to a decision block 2306 where the input device ID 1804 is compared with the list of input device IDs issued by the particular input device distributor. If the input device ID 1804 is validated, flow moves out the "Y" path to a function block 2308 where the message packet is appended with the input device ID 1804 and distributor routing address. Flow then moves to a decision block 2310 where the ARS 308 determines if any E-commerce information is to be associated with a particular input device ID 1804. If so, flow is out the "Y" path to a function block 2312 where the message packet is appended with the E-commerce routing string. The E-commerce routing string provides addressing for the E-commerce server 1618. Flow then moves to a function block 2314 where all message packets are returned back to the PC 302 for processing.

Referring back to decision block 2306, if the input device ID 1804 is determined to be invalid, flow moves out the "N" path and jumps forward to the input of decision block 2314, since the lack of a input device ID 1804 interrupts the link to any advertising provided by the E-commerce server 1618. At this point, the only information provided is the link to the advertiser server 312 for return of product information. Referring now to decision block 2310, if no E-commerce information is available, flow moves out the "N" path and jumps forward to the input of function block 2314 where the message packet back to the PC 302 contains only the URL of the advertiser server 312, the bar code information, the distributor server 1616 address and input device ID 1804 information.

Referring now to FIGURE 24, there is illustrated a flowchart of the process performed at the E-commerce site. The E-commerce server 1618 receives the message packet from the user PC 302, as indicated in function block 2400, and decodes the packet to perform a match with the bar coded information. Moving on to a decision block 2402, if the

match is unsuccessful, flow is out the "N" path to a function block 2404 where the match is rejected. A message may be returned to indicate that a problem occurred and the user may need to re-scan the product bar code 1606. If a successful match occurs, flow moves out the "Y" path to a function block 2406 where the input device ID 1804 is matched with the bar code product information. The bar coded information may be distributed to customers over a large geographic area. However, the input device 1606 may be coded for certain geographic areas. For example, a input device 1600 having an XXX ID may be restricted for sale in the Southwestern United States while a input device 1600 having a YYY ID may be sold only in the Northeast. In this way, geographic areas may be targeted with advertising more appealing to that particular area. Advertising returned to the user PC 302 may be focused further by obtaining a user profile when the software or input device 1600 are installed. In this way, advertising may be focused based upon the user profile. Therefore, flow moves to a function block 2408 to lookup the E-commerce action based upon the input device ID 1804 and the bar code information. Flow moves to a function block 2410 to assemble all the information into a packet for return to the user PC 302. The product information and/or user profile information may be returned. Flow is then to a function block 2412 where the message packet is transmitted.

Referring now to FIGURE 25, there is illustrated a general block diagram of another disclosed embodiment relating to a method of accessing one or more remote locations on a network by sensing a machine-resolvable code. The computer (or "PC") 302 is disposed on a network by means of a suitable interface 304. The network may be any type of computer network including a global communications network 306 such as the Internet. The PC 302 is operably connected to an input device 2502 having a sensor 2504 suitable for resolving at least one type of machine-resolvable code 2506. The input device 2502 may be connected directly to the PC 302 or connected through a suitable interface 2507. The interface 2507 may serve more than one device, for examine, FIGURE 25 shows interface 2507 connecting both input device 2502 and keyboard 1610 to the PC 302. Running on the PC 302 is a software application 2520 which includes a software identification code 2522. In response to sensing the machine resolvable code 2506 with the input device 2502, the PC 302 accesses, via the GCN 306, at least one remote location 2524 associated with the software

identification code 2522. The remote locations accessed may then return remote information to the PC 302 for presentation on the display 1612, for storage on the local database 1614, or for other processing.

The machine-resolvable code 2506 may have the form of any type of information which can be sensed and interpreted by a machine, including a bar code 2508, text 2510 suitable for optical character recognition ("OCR"), audio signals 2512 (regardless of whether they are human-audible), optical signals 2514, magnetic signals encoded on a magnetic stripe 2516 such as found on a credit card, or other electromagnetic signals such as those from an induction coupled transceiver device 2518. The machine resolvable code 2506 sensed by the input device 2502 is generally associated with a product, service or event for which information is available from remote sites on the network. For example, the bar code 2508 may be a product bar code, such as a UPC, which is marked directly on a product or package for use in retailing. The bar code could also be of a non-standard type for providing information regarding an advertisement or identifying items displayed in a newspaper periodical or advertisement. The text 2510 may be human readable text which is also machine readable through optical character recognition. The audio signal 2512 may originate from an on-air television or radio program, it may be embedded in the audio track of a video tape, CD Rom, DVD, or cassette. The optical signal 2514 may be a region of modulated brightness on a television screen resulting from a signal received from an on-air television broadcast, or embedded in the video portion of a video tape, CD Rom or DVD. The magnetic information 2516 may be encoded on a credit card or on other types of cards such as those used for identification or access to remote areas. The induction coupled card 2518 could be of the type used for contact-free communication such as between the card and an electromagnetic scanner as might be used in access control systems. The current embodiment utilizes the sensing of the machine-resolvable code 2506, regardless of its exact type and regardless of whether it was originally intended to have any network significance, to cause the computer 302 to access a remote site over the network.

As previously discussed, a software application 2520 runs on the PC 302. The software application 2520 includes a software identification code 2522 associated with at

least one remote site 2524 on the network 306. The associated remote site 2524 (hereinafter the "software ID site") may include information relevant to the software application, or it may include advertising content particularly selected for presentation to users of the software 2520. By use of this embodiment, the computer 302 can access remote sites selected, at least in part, on the basis of an association with the software ID code 2522.

The machine-resolvable code 2506 may be associated with at least a second remote site on the network 306, for example an advertiser server 312. The second remote site may include information relevant to the machine-resolvable code 2506, or it may include advertising content particularly selected for presentation to users in proximity to the machine-resolvable code 2506. By use of further embodiments, the computer 302 can access remote sites selected on the basis of association with the software ID 2522 and the machine-resolvable code 2506.

In operation, a machine-resolvable code ("M-R code") 2506 is sensed by the input device 2502 and then passed to the PC 302. The sensing of the M-R code 2506 may result from an active user activity, such as scanning a bar code or credit card, or it may result from a passive activity at the user's site, such as receiving a TV, radio or other media signal containing an embedded code. In response to the PC 302 receiving the M-R code 2506, the application program 2520 running on the PC converts the software identification code 2522 and generates routing information for transmission to a second computer on the network, e.g., an ARS 308 as previously described.

Referring now FIGURE 26, there is illustrated a sample message packet transmitted from the user's PC 302 to the ARS 308. The message packet 2602 comprises a number of bits of information including the machine-resolvable code 2506 sensed by the input device 2502; the software identification code 2522 which is derived from the software application 2520; and a user ID 1806 which is derived from the software running on the PC 302 and which identifies uniquely with the user's location. Note that the message packet includes other information necessary for the proper transmission of data from point to point across the GCN 306.

Referring now to FIGURE 27, there is illustrated another sample message packet which can be transmitted from the user's PC 302 to the ARS 308. The message packet 2702 comprises a number of bits of information including an input device ID 1804 which is embedded in a memory in the input device 2502 and identifies it with a particular input device distributor, as well as the machine resolvable code 2506, software ID code 2522 and user ID 1806 comprising the message packet previously discussed.

Referring now to FIGURE 28, there is illustrated a more detailed block diagram of the routing of the message packets in order to present the accessed information to the user. As is mentioned hereinabove, when the input device 2502 senses a machine resolvable code 2506, a software program running on the user PC 302 is operable to interpret the information 2802 output by the input device 2502 and generate a message packet for transmission over the GCN 306. The software program assembles the message packet such that it is directed to the ARS 308 disposed on the GCN 306. As seen in FIGUREs 26 and 27, the message packet contains several pieces of information including the software identification code 2522 which associates it with at least one remote site, machine-resolvable code 2506 which associates it with at least a second remote site (e.g., relating to a particular product of interest to the user or to the sponsor of the code), possibly also an input device ID 1804 which links the input device 2502 to the input device distributor and a user ID 1806 which identifies the network location of the user's computer. This message packet from the PC 302 is transmitted over a path 1900 to the ARS 308 where the ARS database 310 is accessed to cross reference the software ID information 2522 to the associated first site (the "software ID site"), the M-R code 2506 to the associated second site (e.g., the advertisers server), and the input device ID 1804 to yet another remote site (e.g., the input device distributor site). The ARS 308 returns a message packet over a path 1902 to the user PC 302 which contains routing information as to the various other sites disposed on the GCN 306, for example, the software ID site 2524, the advertiser server 312, input device distributor site 1616, and the E-commerce site 1618.

It can be appreciated that other information can also be provided by the ARS 308 which more closely targets the particular user sensing the M-R code 2506. For example, if

it is known that software having a particular software code 2522 is sold in a certain geographic area, this information can be useful in targeting the particular user with certain advertising information relevant to that geographical area. In any case, the information returned from the ARS 308 over the path 1902 provides enough information for the input device program running on the user PC 302 to identify a number of remote sites disposed on the GCN 306. The user PC 302 then processes the return message packet and routes another message packet over a path 2810 to the software ID site 2524. The software ID site 2524 then returns information associated with the particular software identification code 2522 which was running on the user PC 302 over a path 2812. Similarly, the user PC 302 routes information (e.g., the URL of the site relating to the M-R code and user profile) to the advertiser server 312 over path 1904 in order to obtain information back over a path 1906 for framing any banners which relate to the sensed M-R code 2506. Additionally, the user PC 302 may forward a message packet to the input device distributor site 1616 over a path 1908 in order to obtain information back over a path 1910 for framing any banners which identify the input device distributor and may also forward a message packet to the E-commerce site 1618 over a path 1912 in order to return information regarding any particular advertisements to be displayed to the user. Such advertisements are returned to the PC 302 over a path 1914.

Referring now to FIGURE 29, there is illustrated a block diagram of a browser window according to another disclosed embodiment. The browser window 2900 is partitioned into a plurality of areas for framing specific information. A M-R code-specific area 2902 displays information relevant to the M-R code 2506, whose sensing activated the user's computer; a software ID-specific area 2904 displays information associated with the software ID 2522 of the software on the user's computer; an input device-specific area 2004 displays information about the input device distributor; and an E-commerce area 2006 displays other advertising information, for example, information that the input device distributor selects for display according to this particular user and input device. As mentioned hereinabove, a software program operable in response to the sensing of M-R code 2506 processes the software ID code 2522 and develops the browser the window by partitioning it into specific areas for the framing of information. Therefore, information

returned from the E-commerce site 1618 is passed through the GCN 306 to the particular E-commerce frame 2006. Similarly, information relevant to the M-R code 2506 is returned from the advertiser site 312 across the GCN 306 to the particular M-R code-specific area 2902. Information placed in the input device-specific area 2004 is information about the input device distributor which is returned from the input device distributor site 1616 across GCN 306. Of course, information placed in the software ID-specific area 2904 is information associated with the software ID 2522 which is returned from the software ID site 2524 across GCN 306.

Referring now to FIGURE 30, there is illustrated another structure of information which may be contained in the ARS database. As previously described, the ARS database 310 contains a varicty of information required to properly interrogate and assemble packets for obtaining information from the various remote sites disposed on the GCN 306. The ARS database 310 may have a database structure 3000 which contains addresses or other routing information for the remote sites containing information relevant to codes sent by the user's computer or advertising content specifically selected for presentation to a computer utilizing such codes. Under a SOFTWARE heading 3004, are listed the particular software ID codes and routing information for addressing the associated first remote sites for each respective software ID code. For example, the ARS server 308 may contain any number of advertisers having unique URL addresses associated therewith. Therefore, the software ID code 2522 of a particular software application is associated with a URL address which facilitates the accessing of the desired remote sites based on the software ID code. Also part of the ARS database structure 3000 is a heading M-R CODE 3002 under which is the M-R code whose sensing initiated the action and routing information associated with a second remote site relevant to the respective M-R codes. The ARS database structure 3000 may also contain a USER heading 3006 containing information comprising a user's transaction profile and/or an E-COMMERCE heading 3008 containing codes and routing information for use as previously described.

Referring now to FIGURE 31, there is illustrated a flowchart of the process of receiving information from the user's perspective, and according to another disclosed

embodiment. The software application 2520 running on the user's PC 302 may run in the background until activated by the output from the input device 2502. Therefore, flow moves to a decision block 3100, where if an M-R code has not been sensed, flow moves out the "N" path and loops back to the input of decision block 3100. On the other hand, if an M-R code 2506 has been sensed by the input device 2502, flow moves out the "Y" path to a function block 3102, where the application software 2520 assembles a message packet containing the M-R code 2506, the software ID code 2522 and the URL address for the ARS 308. Additionally, a browser is launched in which this information is placed for transmission to the ARS 308. Flow then moves to a function block 3104 where the browser partitions the screen into any number of areas ("frames") in which information is displayed when obtained from the various remote sites, for example, from the advertiser server 312 and from the software ID site 2524. It should be known that although four frames are shown in the particular window 2900 of this embodiment, the number of frames displayed in the window 2900 may be greater or less depending upon the desired effect.

After the software application partitions the browser window into one or more frames in preparation for receipt of return information, flow moves to a decision block 3106 where the computer waits for information to be returned from the ARS 308 disposed on the GCN 306. If the information is not returned, flow moves out the "N" path and simply loops back to the input to continue monitoring for receipt of the information. If information has been received, flows moves out the "Y" path to a function block 3108 where routing information corresponding to each frame (or partitioned area of the window 2900) is inserted into one or more packets for transmission to the various sites on the GCN 306. The various sites then return the requested information back to the PC 302, as indicated in function block 3110. Flow is then to a function block 3112, where the software application 2520 working in conjunction with the browser places the returned information into the respective frames of the window 2900. The user, viewing the display at PC 302, then perceives a variety of information, at least a portion of which is information associated with the software ID 2522, and possibly other information or advertisements based upon the M-R code 2506, the input device ID 1804 and possibly other advertisements based on the user's profile.

Referring now to FIGURE 32, there is illustrated a flowchart of the process of the current embodiment according to the ARS. The ARS 308 is operable to decode and process messages received from the GCN 306. Therefore, flow is to a decision block 3200 where, if a message is not received, flow is out the "N" path with loop back to its input. If a message has been received, flow is to a function block 3202 where the message is analyzed to see if a software ID code 2522 is included. If no software ID code received, flow is out the "N" path to decision block 3208 for further processing as described below. If a software ID code is included in the message, flow is to a function block 3204 where the ARS database 310 is consulted to match the software ID code 2522 to at least one associated remote site 2524. Flow then moves to function blocks 3206 where the routing string for the software ID site 2524 is assembled. Flow then continues to a decision block 3208 where the message is analyzed to see if additional codes have been received, for example, M-R codes 2506, or input device ID code 1804. If such additional codes are present, the ARS database 310 is again consulted to match the additional codes to associated remote sites and assemble the corresponding routing strings. Once these additional codes have been assembled, or if the output from decision block 3208 follows the "N" path, flow continues to function block 3212 where all routing strings are returned from the ARS 308 to the PC 302 using information from the user ID 1806.

Referring now to FIGURE 33, there is illustrated a flowchart of the process performed at the software ID site. The software ID server 2524 receives the message packet from the user PC 302, as indicated in the function block 3300. This message packet was routed to the software ID server 2524 using the routing string information returned from the ARS 308. Moving on to a function block 3302, the decoded message packet is used to assemble information from the server 2524 which is associated with the software ID code 2522 into a new message packet including information for routing the package back to the user's PC 302. Flow then continues to function block 3304 where the software ID server 2524 transmits the message packet including the information associated with the software ID 2522 back to the user PC 302.

Although the preferred embodiment has been described in detail, it should be understood that various changes, substitutions and alterations can be made therein without departing from the spirit and scope of the invention as defined by the appended claims.

WHAT IS CLAIMED IS:

1. A method of accessing one or more remote locations on a network by sensing a machine-resolvable code, comprising the steps of:

providing a first computer disposed on the network, the first computer being connected to an input device for sensing a machine-resolvable code present at a first location, the first computer running a software application which includes a software identification code having associated therewith at least one of the one or more remote locations;

accessing with the first computer a second computer disposed on the network in response to sensing the machine-resolvable code present at the first location with the input device;

performing a lookup operation at the second computer to match the software identification code with the associated at least one of the one or more remote locations to obtain remote routing information corresponding to the associated at least one of the one or more remote locations;

returning the remote routing information corresponding to the associated at least one of the one or more remote locations from the second computer to the first computer; and

accessing with the first computer the associated at least one of the one or more remote locations according to the remote routing information to retrieve remote information.

2. The method of Claim 1, wherein the step of accessing further comprises the steps of:

returning information from the associated at least one of the one or more remote locations to the first computer; and

presenting at least a portion of the information so returned on the display of the first computer for presentation to the user.

3. The method of Claim 1, wherein in response to the sensing of a machine-resolvable code using the input device, the software application running on the first computer converts the software identification code and generates routing information for transmission to the second computer.

4. The method of Claim 3, wherein the routing information includes the software identification code and the address of the second computer.

5. The method of Claim 1, wherein the machine-resolvable code is an optical code and the input device is an optical code scanner.

6. The method of Claim 5, wherein the optical code is a bar code and the optical code scanner is a bar code scanner.

7. The method of Claim 6, wherein the bar code is a universal product code (UPC) bar code.

8. The method of Claim 5, wherein the optical code is alphanumeric text and the optical code scanner is an optical character recognition (OCR) scanner.

9. The method of Claim 5, wherein the optical code is a portion of a display screen displaying a pattern of modulated brightness and the optical code scanner comprises a light sensor.

10. The method of Claim 1, wherein the machine-resolvable code is an audio tone and the input device comprises a microphone.

11. The method of Claim 1, wherein the machine-resolvable code is a magnetic pattern in a strip of magnetic material and the input device is a magnetic strip reader.

12. The method of Claim 1, wherein the machine-resolvable code is a pattern of electromagnetic signals transmitted from an induction-coupled transceiver device and the input device is an electromagnetic signal receiver.

13. The method of Claim 1, wherein the machine-resolvable code is associated with at least a second of the one or more remote locations and the step of performing a lookup operation at the second computer further comprises matching the machine-resolvable code with the associated at least a second of the one or more remote locations to obtain remote routing information corresponding to the associated at least a second of the one or more remote locations.

14. The method of Claim 13, wherein the step of returning the remote routing information further comprises returning the remote routing information corresponding to the associated at least a second of the one or more remote locations from the second computer to the first computer.

15. The method of Claim 14, wherein the step of accessing further comprises the steps of,

returning information from the associated at least one of the one or more remote locations to the first computer;

returning information from the associated second of the one or more remote locations to the first computer; and

framing at least a portion of the information from the associated at least one of the one or more remote locations and at least a portion of the information from the associated second of the one or more remote locations in a browser window of the first computer for presentation to the user.

16. The method of Claim 1,wherein the network is a global communication network.

17. A system for accessing one or more remote locations on a network by sensing a machine-resolvable code, comprising:

a first computer disposed on the network, the first computer being connected to an input device for sensing a machine-resolvable code present at a first location, the first computer running a software application which includes a software identification code having associated therewith at least one of the one or more remote locations;

a second computer disposed on the network, and accessed in response to the input device sensing the machine-resolvable code present at the first location;

wherein a lookup operation is performed at the second computer to match the software identification code with the associated at least one of the one or more remote locations to obtain remote routing information corresponding to the associated at least one of the one or more remote locations;

wherein the remote routing information corresponding to the associated at least one of the one or more remote locations is returned from the second computer to the first computer; and

wherein the associated at least one of the one or more remote locations are accessed by the first computer according to the remote routing information to retrieve remote information therefrom.

18. The system of Claim 17, wherein at least a portion of the information returned from the associated at least one of the one or more remote locations to the first computer is presented on the display of the first computer.

19. The system of Claim 17, wherein the software application running on the first computer converts the software identification code and generates routing information for transmission to the second computer in response to the sensing of a machine-resolvable code by the input device.

20. The system of Claim 19, wherein the routing information includes the software identification code and the address of the second computer.

21. The system of Claim 17, wherein the machine-resolvable code is an optical code and the input device is an optical code scanner.

22. The system of Claim 21, wherein the optical code is a bar code and the optical code scanner is a bar code scanner.

23. The system of Claim 22, wherein the bar code is a universal product code (UPC) bar code.

24. The system of Claim 21, wherein the optical code is alphanumeric text and the optical code scanner is an optical character recognition (OCR) scanner.

25. The system of Claim 21, wherein the optical code is a portion of a display screen displaying a pattern of modulated brightness and the optical code scanner comprises a light sensor.

26. The system of Claim 17, wherein the machine-resolvable code is an audio tone and the input device comprises a microphone.

27. The system of Claim 17, wherein the machine-resolvable code is a magnetic pattern in a strip of magnetic material and the input device is a magnetic strip reader.

28. The system of Claim 17, wherein the machine-resolvable code is a pattern of electromagnetic signals transmitted from an induction-coupled transceiver device and the input device is an electromagnetic signal receiver.

29. The system of Claim 17, wherein the machine-resolvable code is associated with at least a second of the one or more remote locations; and wherein the second computer performs a lookup operation matching the machine-resolvable code with the associated at

least a second of the one or more remote locations to obtain remote routing information corresponding to the associated at least a second of the one or more remote locations.

30. The method of Claim 29, wherein the second computer returns the remote routing information corresponding to the associated at least a second of the one or more remote locations to the first computer.

31. The method of Claim 30, wherein information from the associated at least one of the one or more remote locations is returned to the first computer; wherein information from the associated second of the one or more remote locations is returned to the first computer; and wherein at least a portion of the information from the associated at least one of the one or more remote locations and at least a portion of the information from the associated second of the one or more remote locations are framed in a browser window of the first computer for presentation to the user.

32. The system of Claim 17,wherein the network is a global communication network.

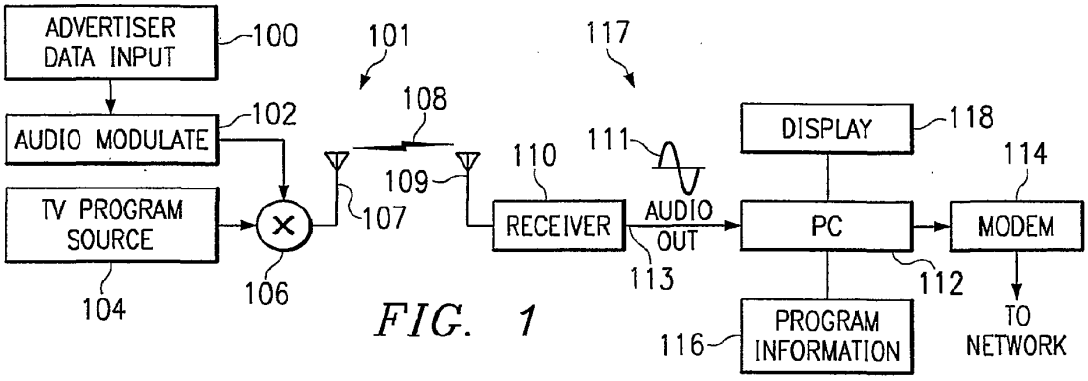


FIG. 1

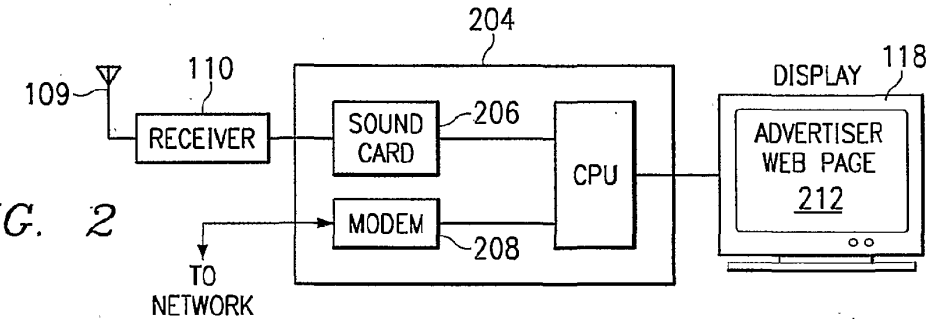


FIG. 2

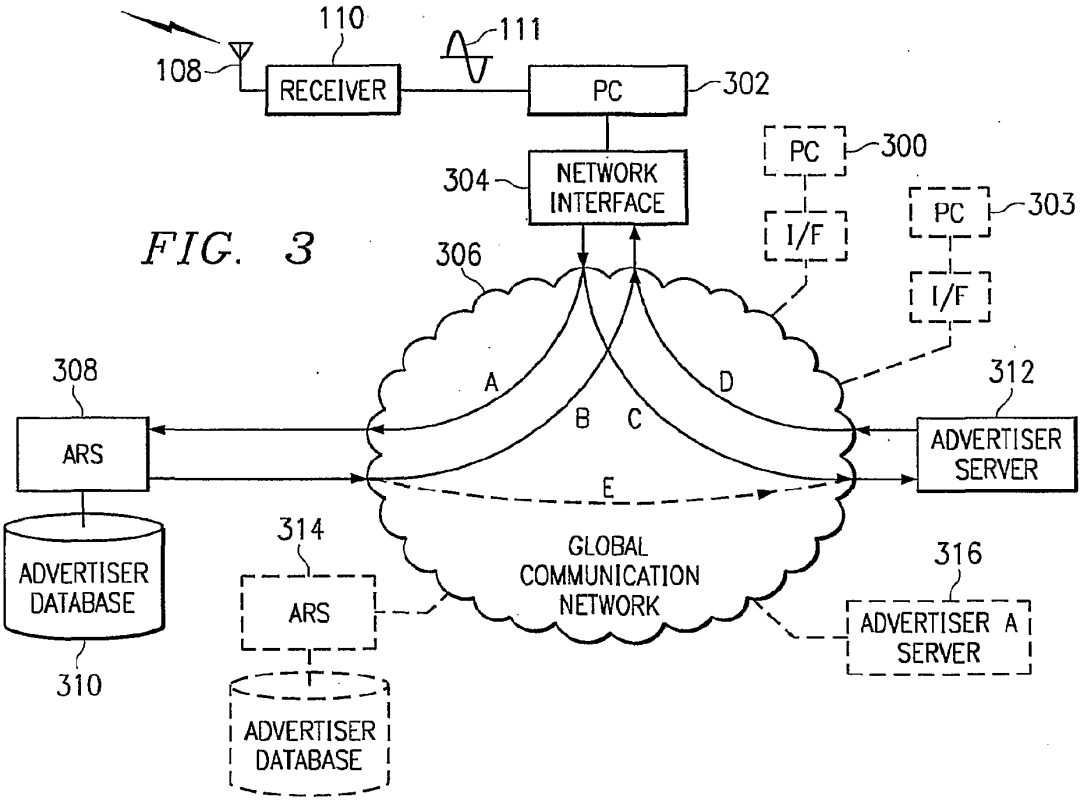


FIG. 3

PATH A: SOURCE TO ARS

400	URL OF ARS	ADVERTISER PRODUCT CODE	URL OF SOURCE

FIG. 4a

PATH B: ARS TO SOURCE

402	URL OF SOURCE	URL OF ADVERTISER	URL OF ARS

FIG. 4b

PATH C: SOURCE TO ADVERTISER

404	URL OF ADVERTISER	REQUEST-FOR-INFORMATION	URL OF SOURCE

FIG. 4c

PATH D: ADVERTISER TO SOURCE

406	URL OF SOURCE	REQUESTED INFORMATION	URL OF ADVERTISER

FIG. 4d

PATH E: ARS TO ADVERTISER (OPTIONAL)

408	URL OF ADVERTISER	REQUEST-FOR-INFORMATION	URL OF SOURCE

FIG. 4e

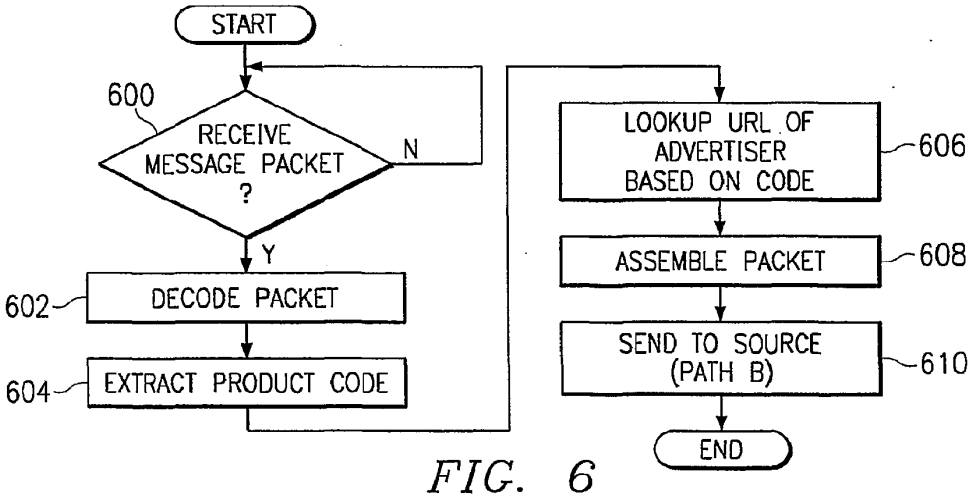


FIG. 6

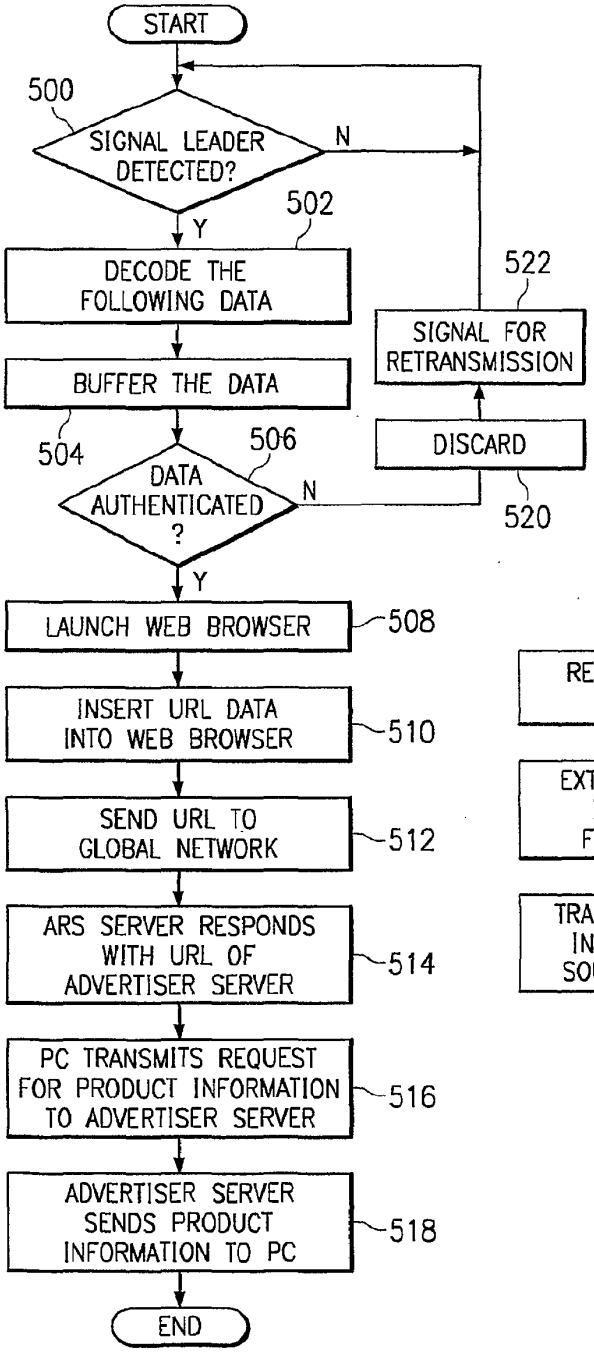

START
500
SIGNAL LEADER DETECTED?
N
Y
502
DECODE THE FOLLOWING DATA
BUFFER THE DATA
504
506
DATA AUTHENTICATED ?
N
Y
522
SIGNAL FOR RETRANSMISSION
DISCARD
520
LAUNCH WEB BROWSER
508
INSERT URL DATA INTO WEB BROWSER
510
SEND URL TO GLOBAL NETWORK
512
ARS SERVER RESPONDS WITH URL OF ADVERTISER SERVER
514
PC TRANSMITS REQUEST FOR PRODUCT INFORMATION TO ADVERTISER SERVER
516
ADVERTISER SERVER SENDS PRODUCT INFORMATION TO PC
518
END

FIG. 5

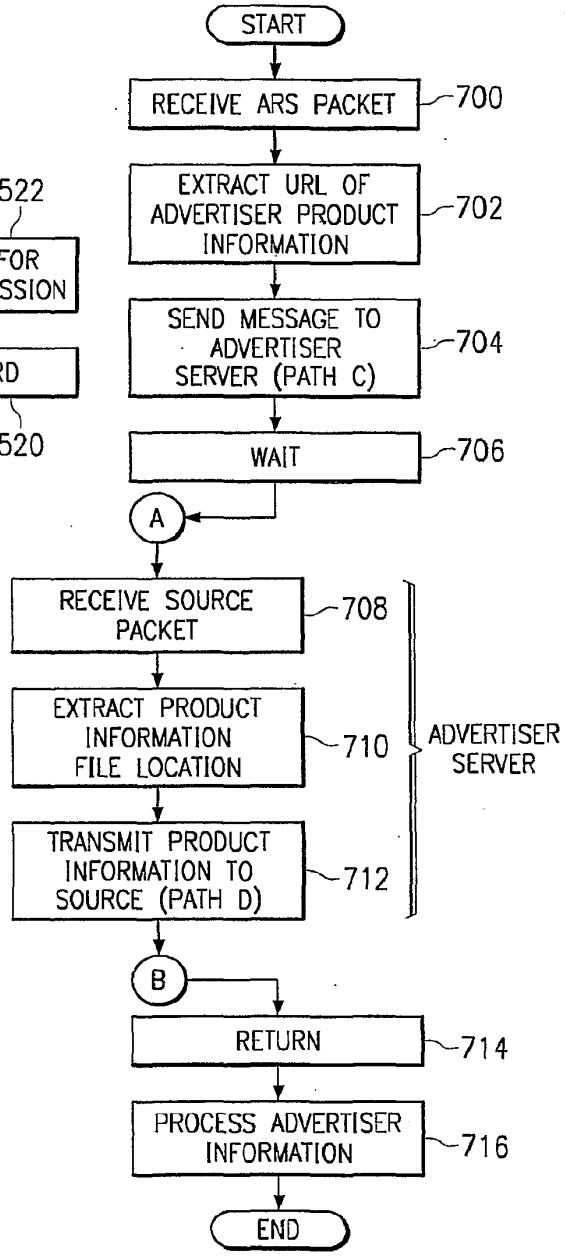

START
RECEIVE ARS PACKET
700
EXTRACT URL OF ADVERTISER PRODUCT INFORMATION
702
SEND MESSAGE TO ADVERTISER SERVER (PATH C)
704
WAIT
706
A
RECEIVE SOURCE PACKET
708
EXTRACT PRODUCT INFORMATION FILE LOCATION
710
TRANSMIT PRODUCT INFORMATION TO SOURCE (PATH D)
712
ADVERTISER SERVER
B
RETURN
714
PROCESS ADVERTISER INFORMATION
716
END

FIG. 7

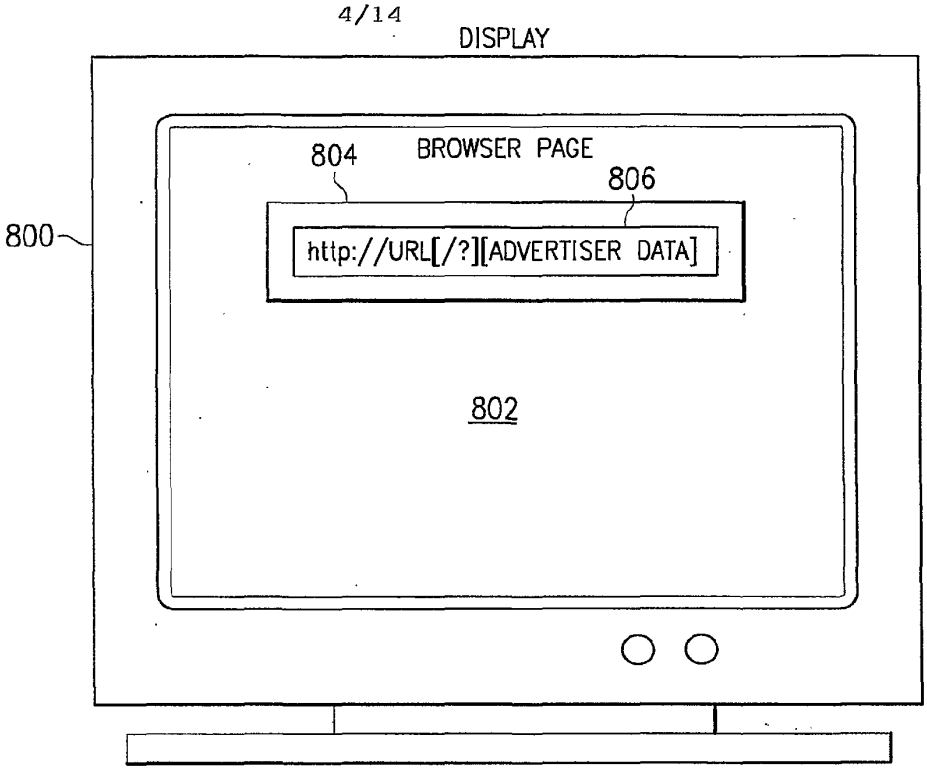
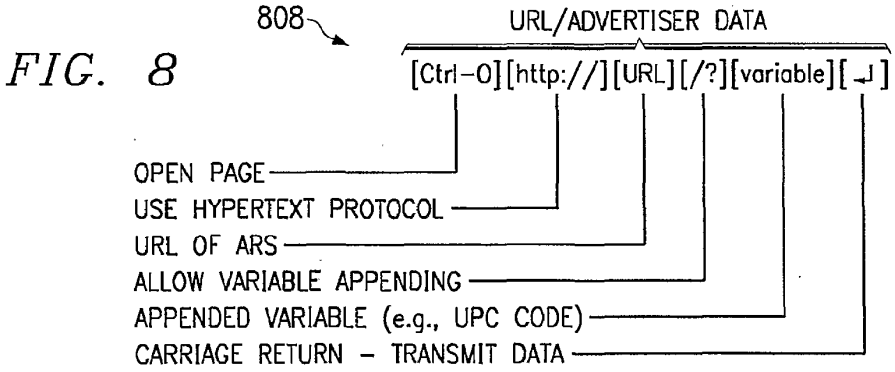

DISPLAY
800
804
BROWSER PAGE
806
http://URL[/?][ADVERTISER DATA]
802

FIG. 8


808
URL/ADVERTISER DATA
[Ctrl-O][http://][URL][/?][variable][↵]
OPEN PAGE
USE HYPERTEXT PROTOCOL
URL OF ARS
ALLOW VARIABLE APPENDING
APPENDED VARIABLE (e.g., UPC CODE)
CARRIAGE RETURN - TRANSMIT DATA

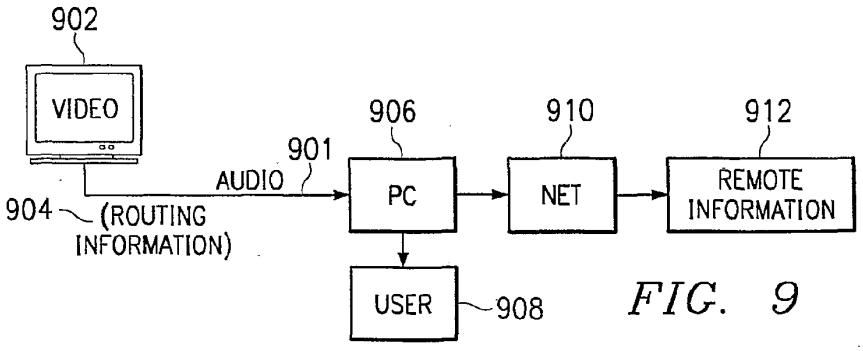

902
VIDEO
906
910
912
901
AUDIO
PC
NET
REMOTE INFORMATION
904
(ROUTING INFORMATION)
USER
908

FIG. 9

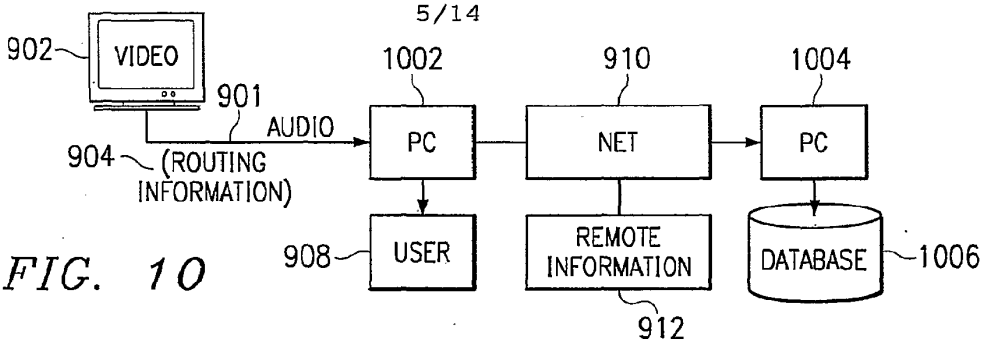


FIG. 10

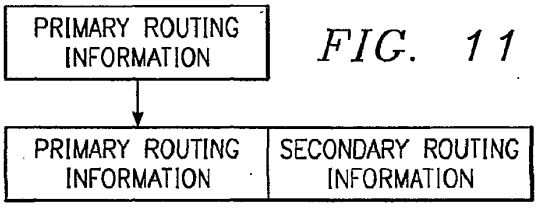


FIG. 11

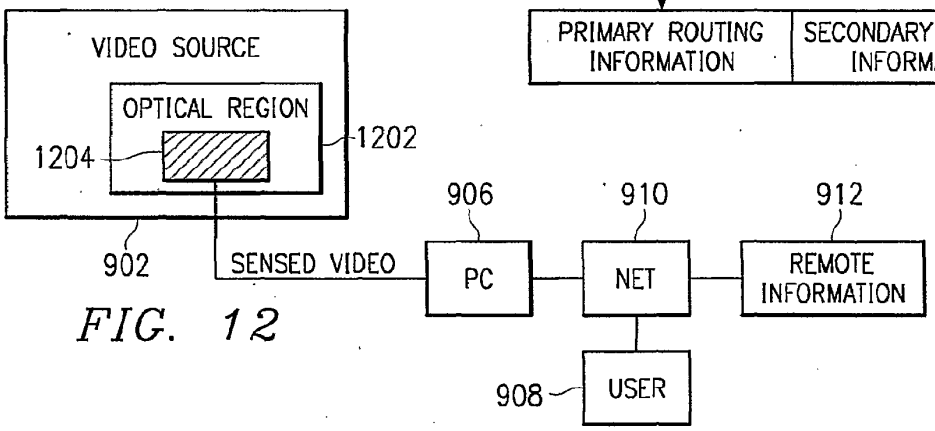


FIG. 12

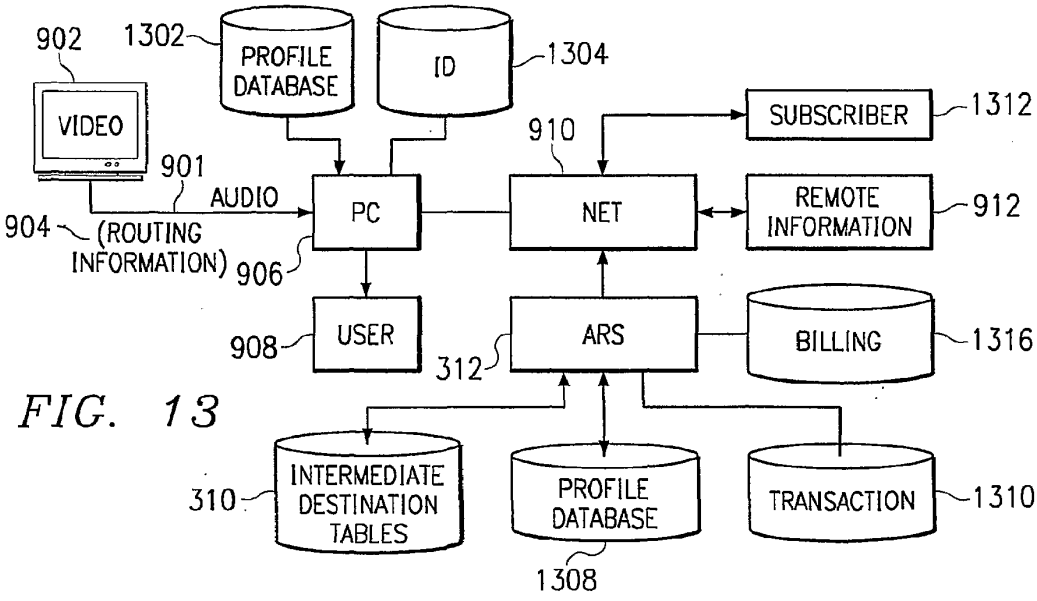


FIG. 13

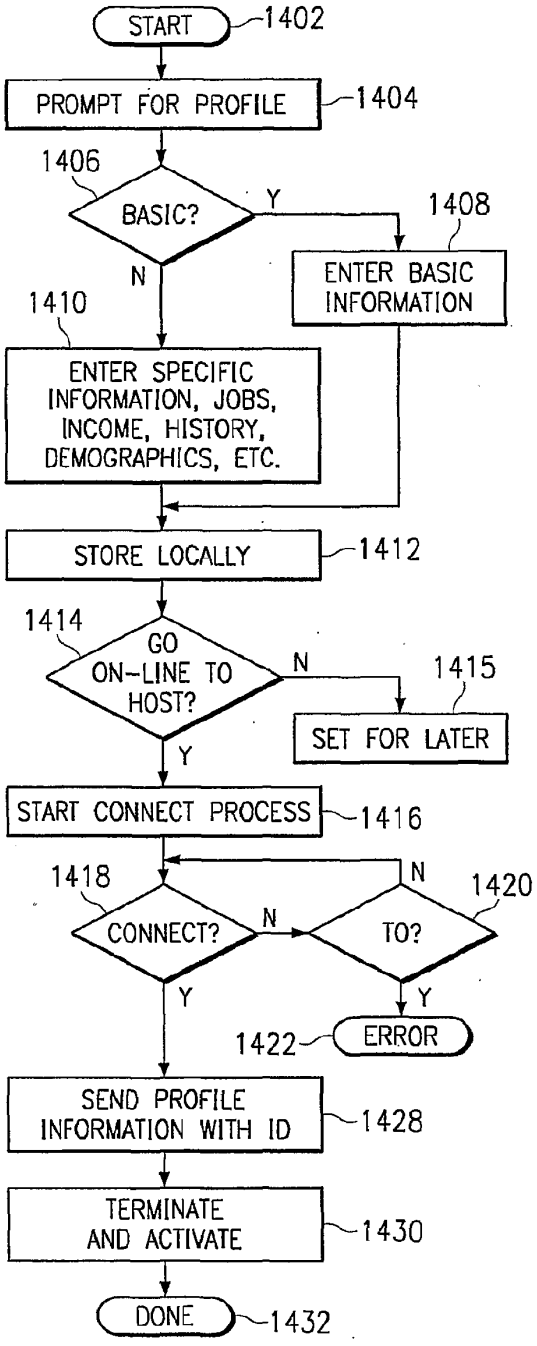

START
1402
PROMPT FOR PROFILE
1404
1406
BASIC?
Y
1408
ENTER BASIC INFORMATION
N
1410
ENTER SPECIFIC INFORMATION, JOBS, INCOME, HISTORY, DEMOGRAPHICS, ETC.
STORE LOCALLY
1412
1414
GO ON-LINE TO HOST?
N
1415
SET FOR LATER
Y
START CONNECT PROCESS
1416
1418
CONNECT?
N
TO?
1420
N
Y
Y
1422
ERROR
SEND PROFILE INFORMATION WITH ID
1428
TERMINATE AND ACTIVATE
1430
DONE
1432

FIG. 14

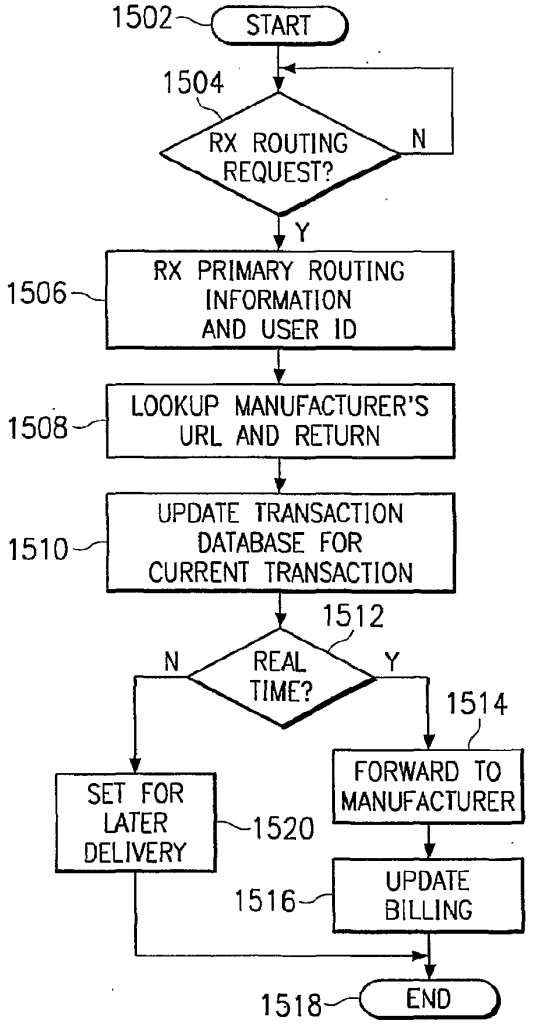

1502
START
1504
RX ROUTING REQUEST?
N
Y
1506
RX PRIMARY ROUTING INFORMATION AND USER ID
1508
LOOKUP MANUFACTURER'S URL AND RETURN
1510
UPDATE TRANSACTION DATABASE FOR CURRENT TRANSACTION
1512
N
REAL TIME?
Y
1514
FORWARD TO MANUFACTURER
SET FOR LATER DELIVERY
1520
1516
UPDATE BILLING
1518
END

FIG. 15

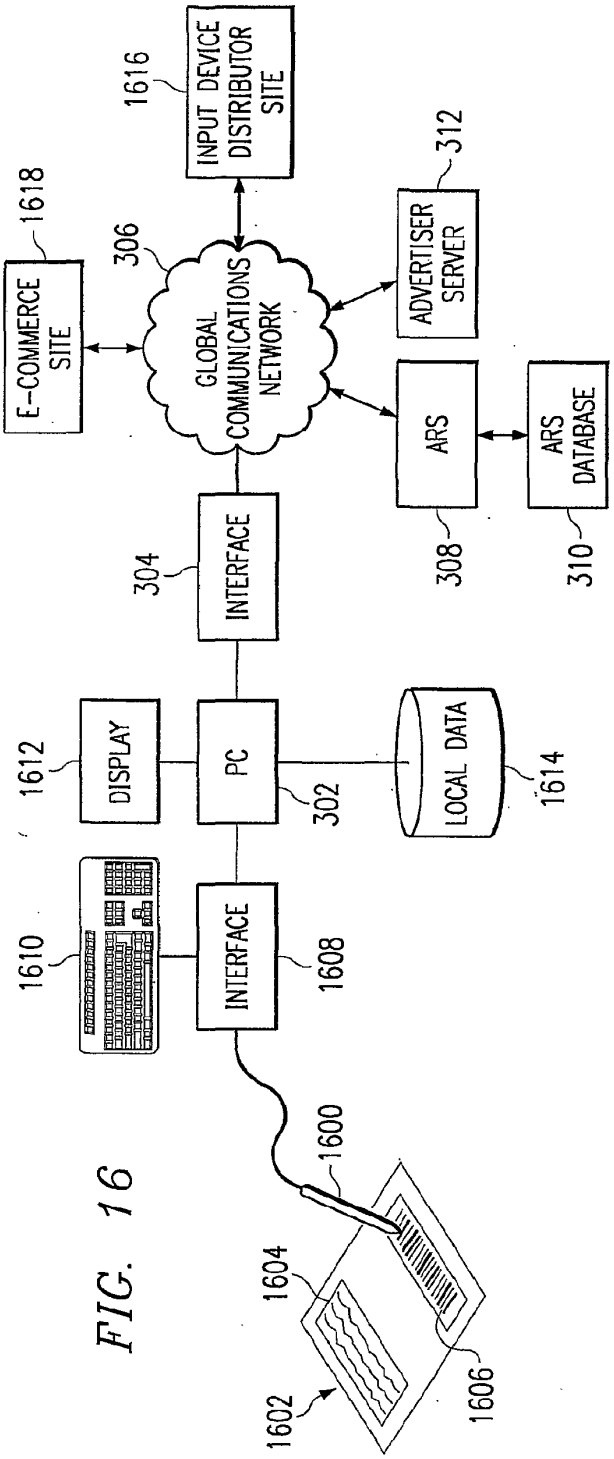


FIG. 16

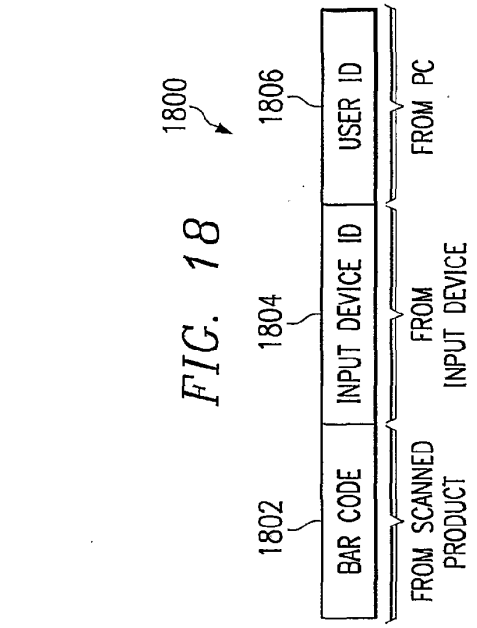


FIG. 18

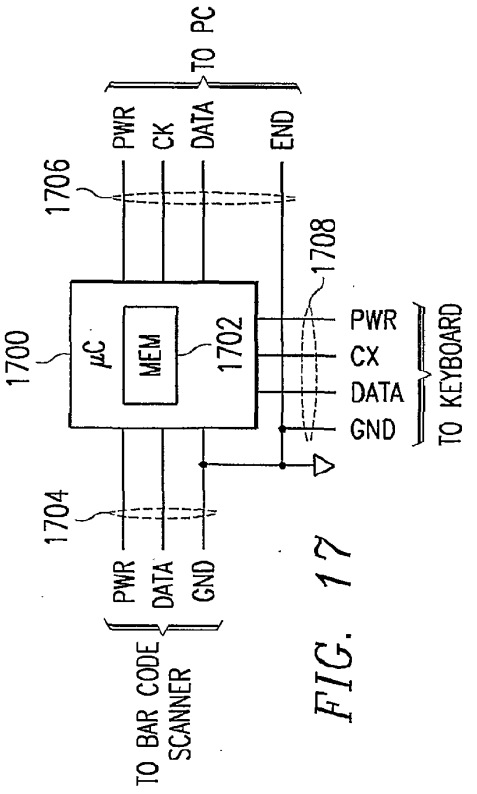


FIG. 17

FIG. 19

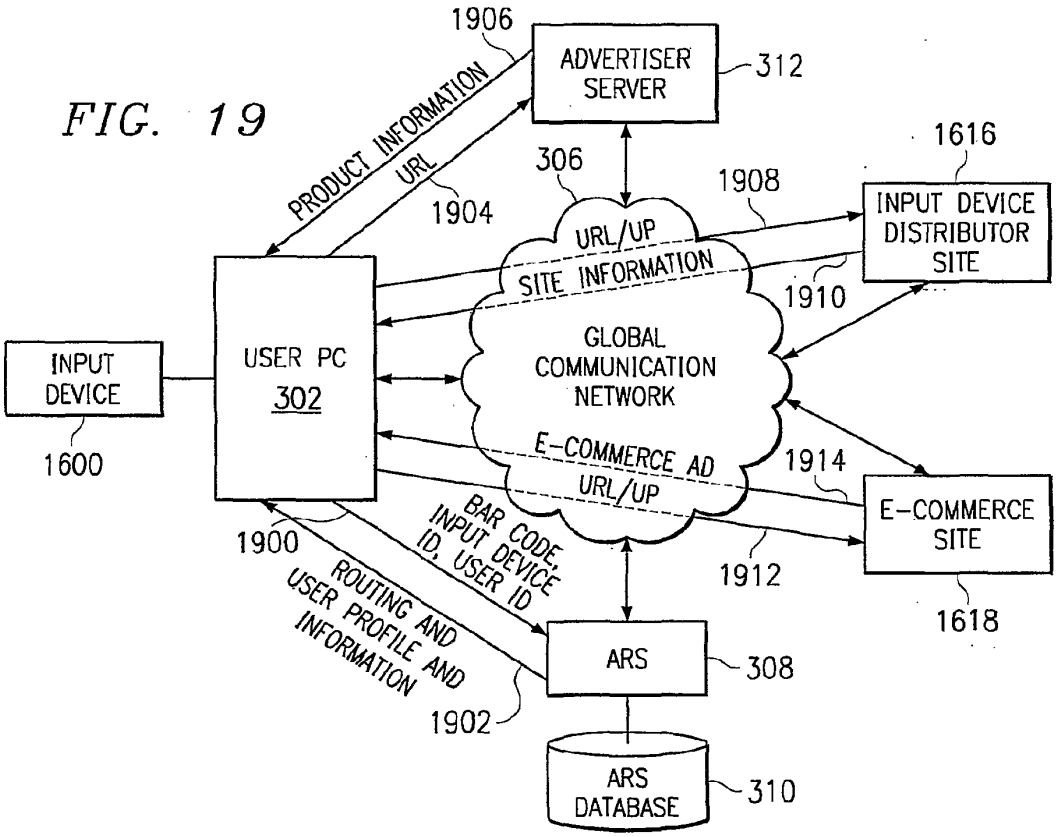

1906
ADVERTISER SERVER
312
PRODUCT INFORMATION
URL
1904
306
1908
1616
INPUT DEVICE DISTRIBUTOR SITE
URL/UP
SITE INFORMATION
1910
INPUT DEVICE
1600
USER PC
302
GLOBAL COMMUNICATION NETWORK
E-COMMERCE AD
URL/UP
1914
E-COMMERCE SITE
1618
1900
BAR CODE, INPUT DEVICE ID, USER ID
1912
ROUTING AND USER PROFILE AND INFORMATION
ARS
308
1902
ARS DATABASE
310

FIG. 20

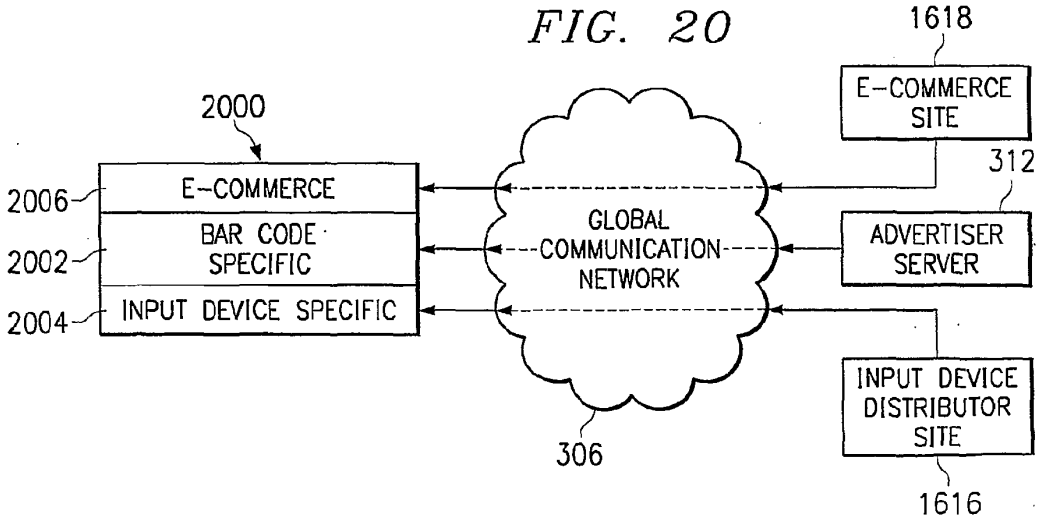

1618
E-COMMERCE SITE
2000
312
2006
E-COMMERCE
2002
BAR CODE SPECIFIC
2004
INPUT DEVICE SPECIFIC
GLOBAL COMMUNICATION NETWORK
ADVERTISER SERVER
INPUT DEVICE DISTRIBUTOR SITE
306
1616

FIG. 21

ARS DATABASE 2100

2102		2104		2106	2108	
PRODUCT		INPUT DEVICE		USER	E-COMMERCE	
BAR CODE	ROUTE	ID	DISTRIBUTOR	PROFILE	BAR CODE	INFORMATION

FIG. 22

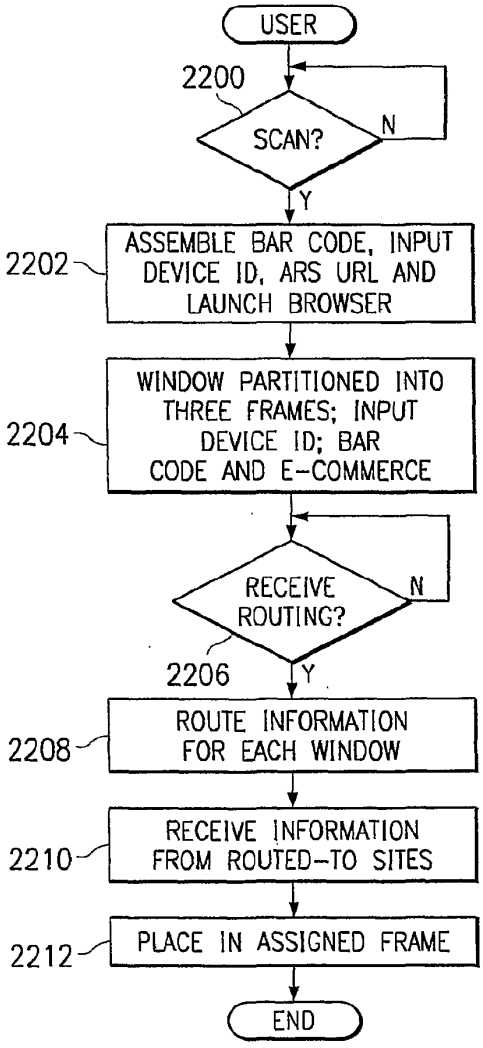


FIG. 23

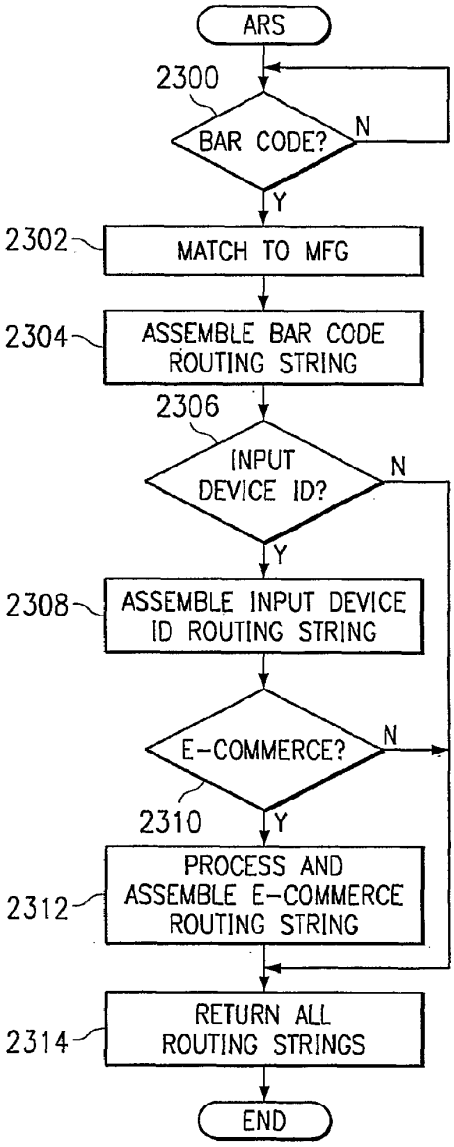
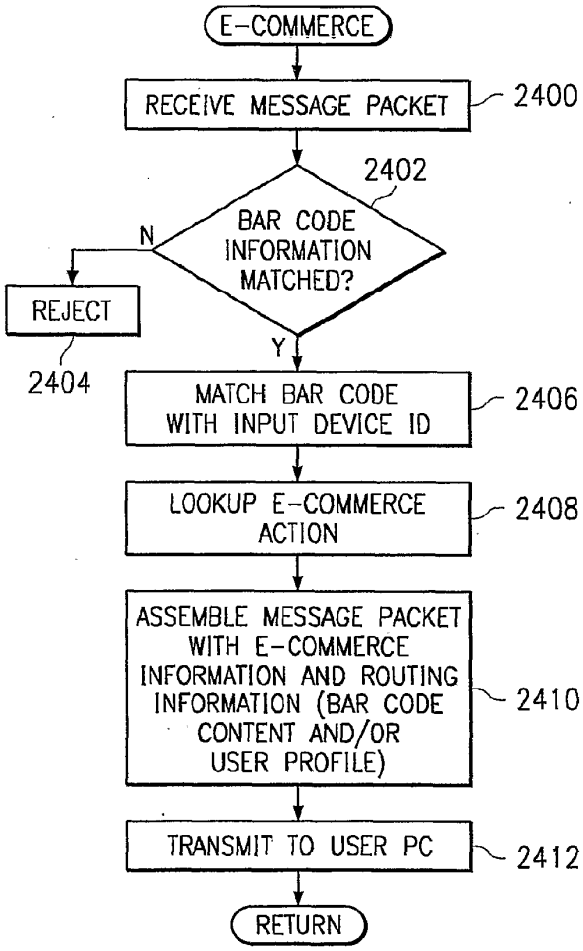

ARS
2300
BAR CODE?
N
Y
2302
MATCH TO MFG
2304
ASSEMBLE BAR CODE ROUTING STRING
2306
INPUT DEVICE ID?
N
Y
2308
ASSEMBLE INPUT DEVICE ID ROUTING STRING
E-COMMERCE?
N
2310
Y
2312
PROCESS AND ASSEMBLE E-COMMERCE ROUTING STRING
2314
RETURN ALL ROUTING STRINGS
END

FIG. 24


E-COMMERCE
RECEIVE MESSAGE PACKET
2400
2402
BAR CODE INFORMATION MATCHED?
N
REJECT
2404
Y
MATCH BAR CODE WITH INPUT DEVICE ID
2406
LOOKUP E-COMMERCE ACTION
2408
ASSEMBLE MESSAGE PACKET WITH E-COMMERCE INFORMATION AND ROUTING INFORMATION (BAR CODE CONTENT AND/OR USER PROFILE)
2410
TRANSMIT TO USER PC
2412
RETURN

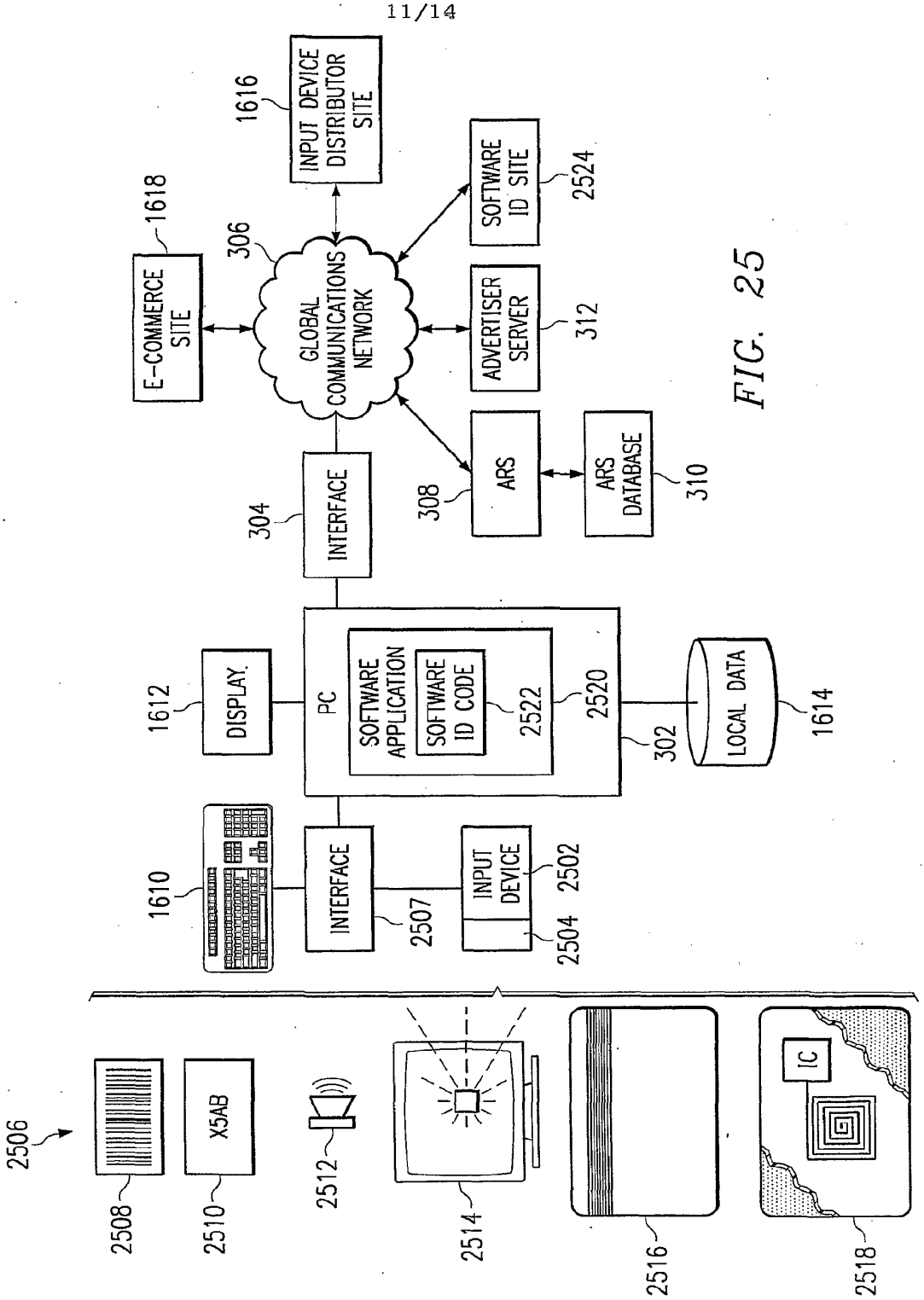

1616
INPUT DEVICE DISTRIBUTOR SITE
1618
E-COMMERCE SITE
306
GLOBAL COMMUNICATIONS NETWORK
SOFTWARE ID SITE
2524
ADVERTISER SERVER
312
304
INTERFACE
308
ARS
ARS DATABASE
310
1612
DISPLAY
PC
SOFTWARE APPLICATION
SOFTWARE ID CODE
2522
2520
302
LOCAL DATA
1614
1610
INTERFACE
2507
INPUT DEVICE
2502
2504
2506
2508
X5AB
2510
2512
2514
2516
IC
2518

FIG. 25

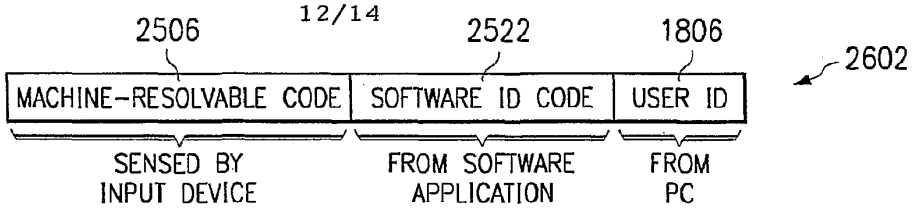


FIG. 26

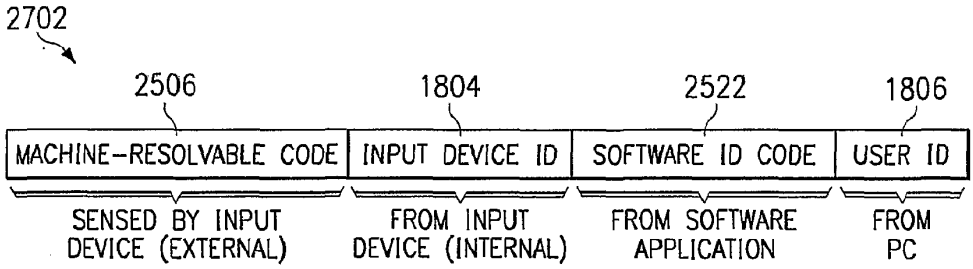


FIG. 27

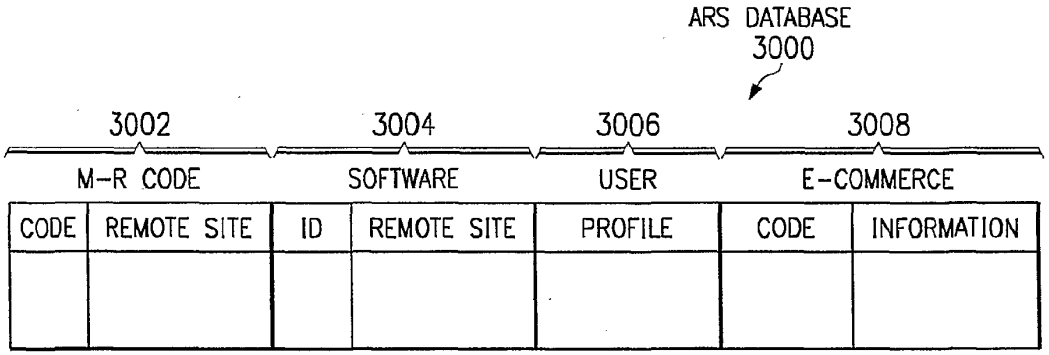


FIG. 30

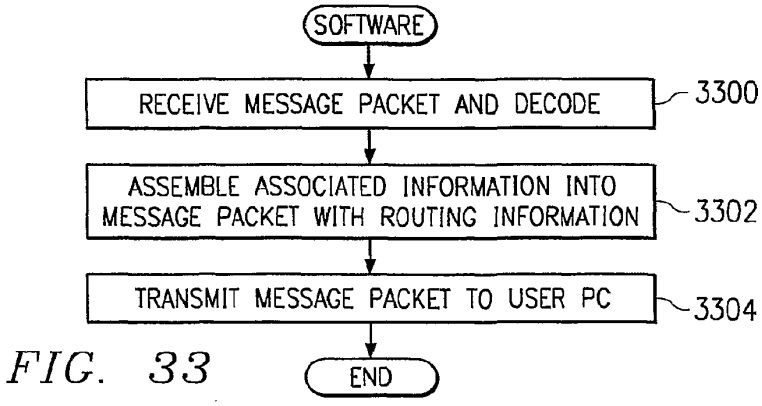


FIG. 33

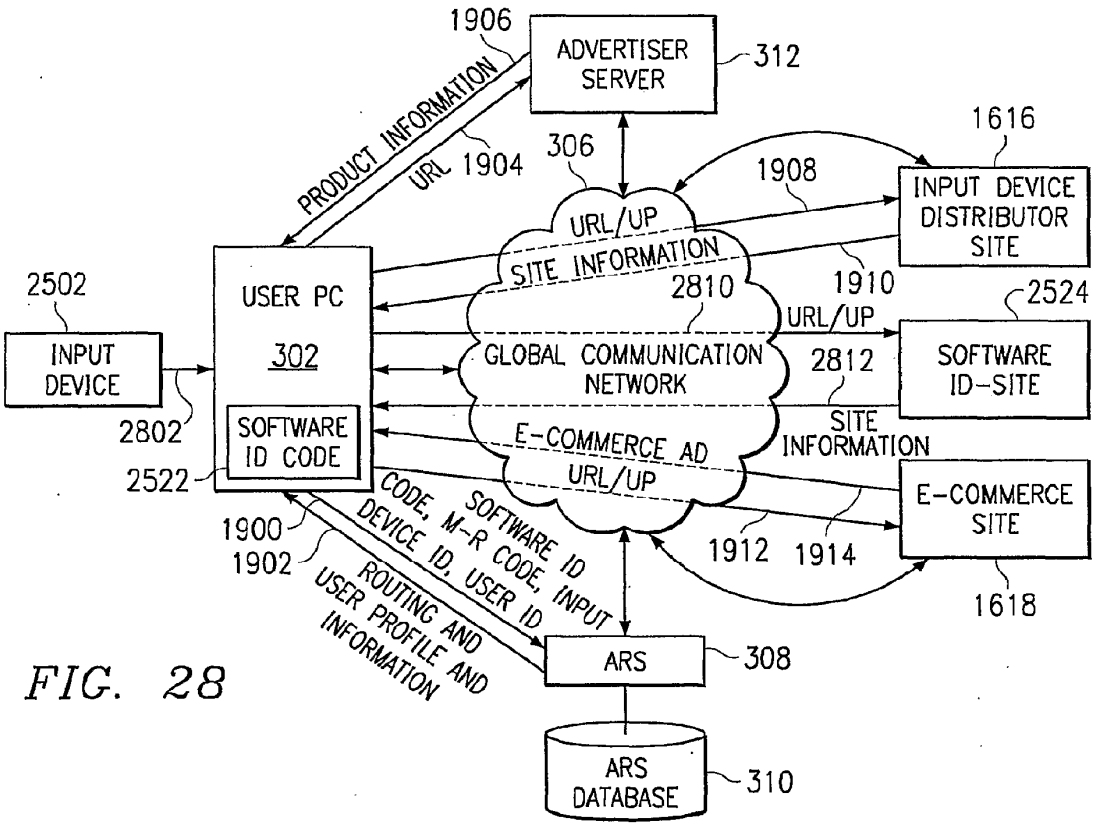

1906
ADVERTISER SERVER
312
PRODUCT INFORMATION
URL 1904
306
1616
1908
INPUT DEVICE DISTRIBUTOR SITE
URL/UP
SITE INFORMATION
2810
1910
2524
2502
USER PC
302
INPUT DEVICE
GLOBAL COMMUNICATION NETWORK
URL/UP
2812
SOFTWARE ID-SITE
2802
SOFTWARE ID CODE
E-COMMERCE AD
SITE INFORMATION
2522
URL/UP
E-COMMERCE SITE
CODE, SOFTWARE ID
DEVICE ID, M-R CODE, USER ID, INPUT
1900
1902
1912
1914
ROUTING AND USER PROFILE AND INFORMATION
1618
ARS
308
ARS DATABASE
310

FIG. 28

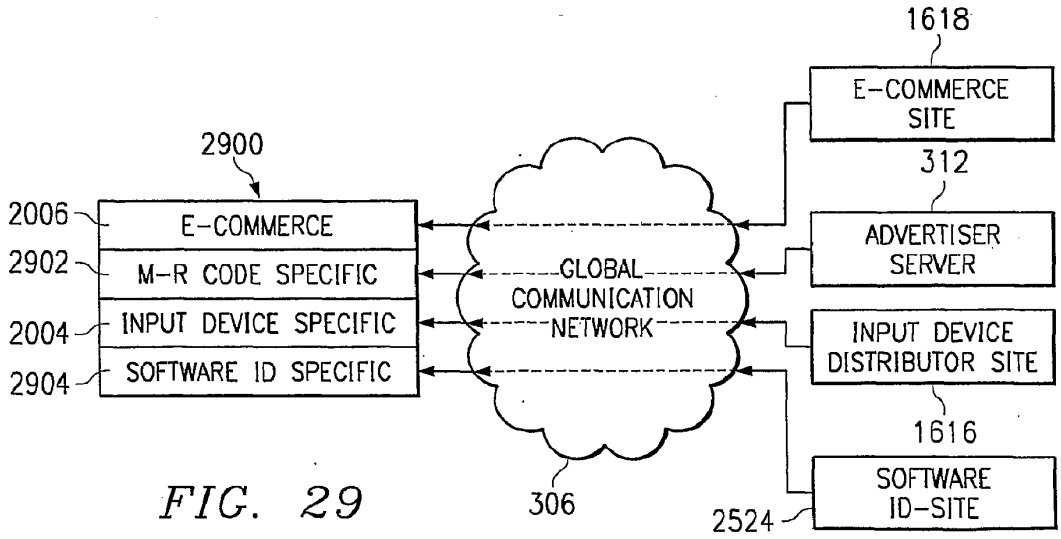

1618
E-COMMERCE SITE
2900
312
2006
E-COMMERCE
ADVERTISER SERVER
2902
M-R CODE SPECIFIC
GLOBAL COMMUNICATION NETWORK
2004
INPUT DEVICE SPECIFIC
INPUT DEVICE DISTRIBUTOR SITE
2904
SOFTWARE ID SPECIFIC
1616
SOFTWARE ID-SITE
306
2524

FIG. 29

FIG. 31

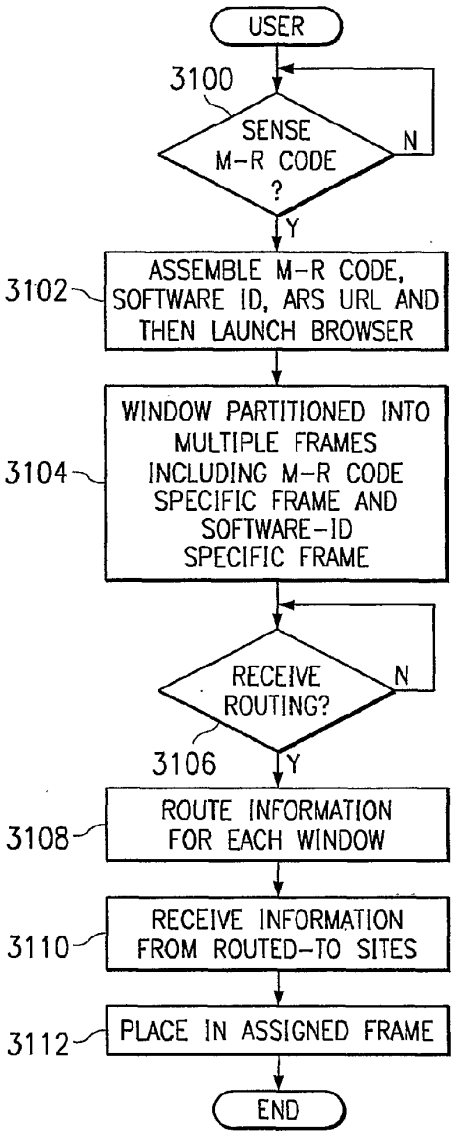
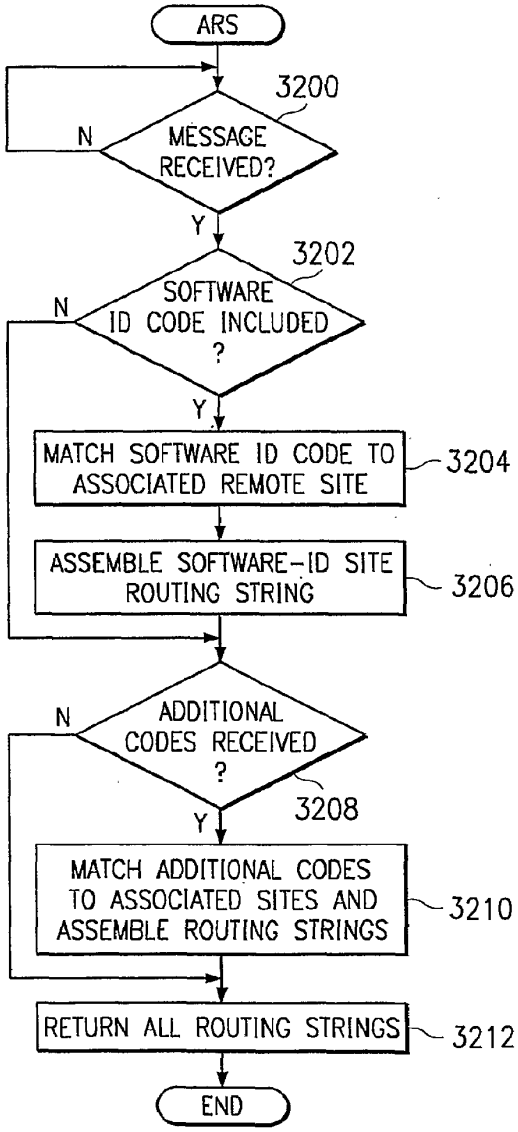

USER
3100
SENSE M-R CODE ?
N
Y
3102
ASSEMBLE M-R CODE, SOFTWARE ID, ARS URL AND THEN LAUNCH BROWSER
3104
WINDOW PARTITIONED INTO MULTIPLE FRAMES INCLUDING M-R CODE SPECIFIC FRAME AND SOFTWARE-ID SPECIFIC FRAME
RECEIVE ROUTING?
N
3106
Y
3108
ROUTE INFORMATION FOR EACH WINDOW
3110
RECEIVE INFORMATION FROM ROUTED-TO SITES
3112
PLACE IN ASSIGNED FRAME
END

FIG. 32


ARS
3200
N
MESSAGE RECEIVED?
Y
3202
N
SOFTWARE ID CODE INCLUDED ?
Y
MATCH SOFTWARE ID CODE TO ASSOCIATED REMOTE SITE
3204
ASSEMBLE SOFTWARE-ID SITE ROUTING STRING
3206
N
ADDITIONAL CODES RECEIVED ?
3208
Y
MATCH ADDITIONAL CODES TO ASSOCIATED SITES AND ASSEMBLE ROUTING STRINGS
3210
RETURN ALL ROUTING STRINGS
3212
END

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 October 2001 (11.10.2001)

PCT

(10) International Publication Number
WO 01/75629 A1

(51) International Patent Classification[7]: G06F 15/16, 15/173

(21) International Application Number: PCT/US01/10336

(22) International Filing Date: 29 March 2001 (29.03.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/193,836	31 March 2000 (31.03.2000)	US
60/193,755	31 March 2000 (31.03.2000)	US
60/193,737	31 March 2000 (31.03.2000)	US

(71) Applicant: NEOMEDIA TECHNOLOGIES, INC. [US/US]; 2201 Second Street, Suite 600, Fort Myers, FL 33901 (US).

(72) Inventors: HUNTER, Kevin, D.; 8381 Arborfield Court, Fort Myers, FL 33912 (US). DURST, Robert, T., Jr.; 6111 Tidewater Island Circle, Fort Myers, FL 33908 (US).

(74) Agent: BARKUME, Anthony, R.; Greenberg Traurig, LLP, 200 Park Avenue, New York, NY 10166 (US).

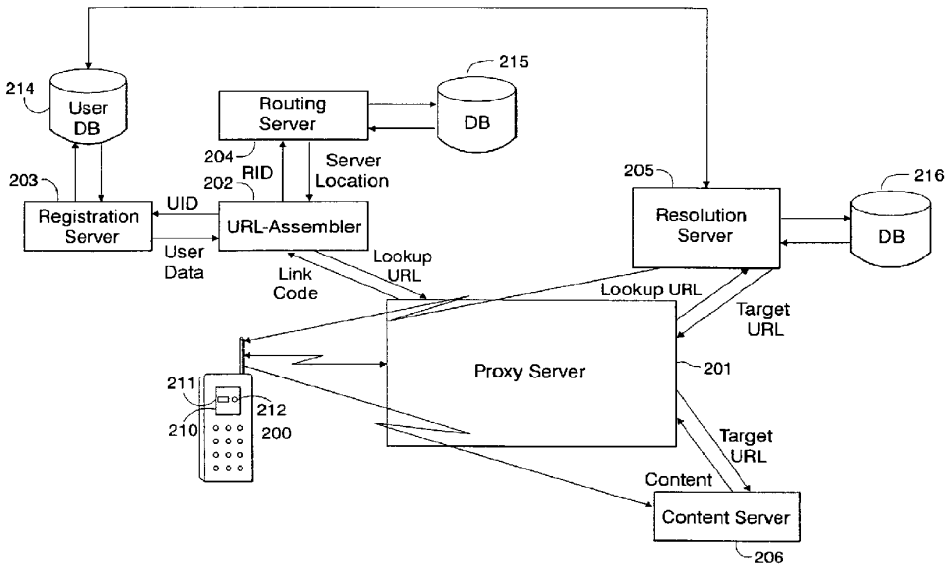
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: SYSTEM FOR ACCESSING INTERNET VIA WIRELESS DEVICE USING LINKAGE URL BAR-CODE



(57) Abstract: A system for accessing a primary content file (206) with a client device (200) that is browser-based (WAP); i.e., that does not require a plug-in type program to be executed on the client device in addition to the browser. URLs based on linkage codes entered into the client device (200) are assembled by a network-based URL-assembly server (202) rather than at the client device, wherein the client device is a wireless device supporting WML content or HTML content.

WO 01/75629 A1

SYSTEM FOR ACCESSING INTERNET VIA WIRELESS DEVICE USING LINKAGE URL BAR-CODE

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority from: (1) Hunter, "METHOD AND SYSTEM FOR SIMPLIFIED ACCESS TO INTERNET CONTENT ON A WIRELESS DEVICE", U.S. Provisional Application No. 60/193,737, filed March 31, 2000, the contents of which are incorporated herein by reference; (2) Hunter, et al., "SYSTEM FOR USING WIRELESS WEB DEVICES TO STORE WEB LINK CODES ON A LIST SERVER FOR SUBSEQUENT RETRIEVAL", U.S. Provisional Application No. 60/193,755, filed March 31, 2000, the contents of which are incorporated herein by reference; and (3) Hunter, "DEVICE-BASED ROUTING FOR WEB CONTENT RETRIEVAL", U.S. Provisional Application No. 60/193,836, filed March 31, 2000, the contents of which are incorporated herein by reference.

TECHNICAL FIELD

This invention relates to a method and system for simplified access of Internet content such as web pages through a wireless device such as a cellular telephone by entry of a linkage code that is associated with such Internet content.

BACKGROUND ART

Recently, a new generation of cell phones has been introduced that include provisions for Internet connectivity and "micro-browsers." Using this class of device, the cell phone user can directly access content on the World Wide Web, receive email, be notified of changes in "subscribed information" such as sports scores or stock prices, etc.

The constraints imposed on a "micro-browser" in a cell phone environment pose a unique problem for both the information provider as well as the user retrieving the

information. The development of the Wireless Application Protocol ("WAP") specification was specifically designed to address a number of fundamental differences between classic Internet and Web-based services and services on a wireless data network. These issues included the differences in needs and expectations as well as differences imposed by the device. That is, wireless devices will generally have less powerful CPU's, less memory and smaller displays than conventional computers. Wireless devices may have very different input devices. Wireless devices other than cell phones may be used that have very different capabilities. All of these issues have been addressed in the WAP specification and architecture. In particular, the WAP system is in no way restricted to cell phones - integration into other devices with wireless connectivity (e.g. Palm Pilots) was clearly anticipated. Thus, although this application will generally refer specifically to cell phones, anything described in that context would also apply to any other comparable wireless device, such as a Personal Digital Assistant ("PDA").

Figure 1 outlines the basic WAP architecture. Wireless devices are not directly "on the Internet" in the same sense as traditional computers. The fact that devices roam around, as well as the sheer number of devices expected to be deployed, discourage a solution in which each wireless device has its own IP address and communicates directly. In addition, the standard Internet protocols require a fair amount of overhead, which poses a problem on a channel with limited bandwidth. As a result, a new set of wireless protocols was developed. These include the Wireless Session Protocol ("WSP") and the Wireless Datagram Protocol ("WDP") which parallel the function of the TCP/IP and UDP Internet protocols. Wireless Telephone Application ("WTA") Servers "speak" these protocols natively, and can be directly accessed by wireless devices.

While for certain applications the requirement of a new class of server is acceptable, restricting wireless devices to this class of server would not adequately leverage the huge embedded base of Internet equipment. As such, the architecture includes WAP proxies, which serve as a bridge between the wireless network and the standard Internet. When a digital device attempts to access a resource via a standard URL, this request is passed to the WAP Proxy using wireless protocols. The WAP Proxy reformats this request into a standard HTTP 1.1 query, retrieves the content from the standard Web Server, and then passes the result back to the wireless device using the wireless protocol. Using this system, wireless devices can access any server on the Internet.

A web site that natively "understood" wireless devices would generally return content in the new Wireless Markup Language (WML), or possibly in the older Handheld Device Markup Language (HDML). Recognizing that achieving deployment of a second, parallel coding standard for documents might slow the penetration of the WAP technology, the WAP Architecture also includes provisions for filters that could translate standard HTML into WML automatically. These filters can be integrated into the WAP Proxy itself, or can exist between the web server and the WAP Proxy. This would, at least in theory, allow a wireless user to access any existing content on the web, even if the web site was not specifically designed for access by devices of this class.

A system and method for utilizing a link code or linkage code to cause a client computer to automatically access a web resource is disclosed in copending U.S. patent application serial number 09/543,178, filed on April 5, 2000 and incorporated by reference herein. In the system

described therein, the link code is a bar code that is scanned by a bar code scanner and input into a client software program that uses the decoded link code to request a URL template from an external server computer. The server takes the link code, returns the URL template, and the link client program assembles the URL using other data at the client. The URL is passed to a web browser, which then retrieves the web resource. This process may also be performed by manually entering a text string associated with the code, such as by entering a UPC number found at the bottom of a typical UPC barcode. This is a powerful way of utilizing a general purpose computer to automatically access a web resource without having to type in a lengthy URL.

It is noted that the system described above relies on the use of a client program running on the client computing device for obtaining, assembling, and then providing the URL to the web browser program. In general purpose personal computers, loading and running of such a client program is of course a typical and easily-done process. It would be desirable, however, to use this content retrieval technology with a client device such as a web-enabled cell phone, in which the use of add-on programs such as the linkage client is not easily accomplished. That is, with the proliferation of web-enabled cell phones and the like, it would be desirable to enable such devices to use such content retrieval methodologies without requiring adaptation of the software or firmware already present on such devices. One aspect of the invention described herein is thus a server-based URL assembly/retrieval methodology that requires no modification to existing web-enabled devices such as web-enabled cell phones or internet kiosks.

Another problem encountered by utilization of web-enabled cell phones for Internet access is that such devices have unique display capabilities. That is, one cannot

simply take any standard HTML web page and display it on a microbrowser or other such device. Thus, it is desired to be able to use the same linkage code in order to provide different types of content to different types of display devices. That is, it is desired to be able to format the content retrieved differently as a function of the device on which it will be displayed. As a result, a user entering a link code with a web-enabled cell phone will be automatically provided with WML or HDML content appropriate for display on that cell phone, while another user entering the same linkage code into a desktop computer will be provided with standard HTML content suitable for display on a large screen.

Although web-enabled cell phones can access web pages using the appropriate linkage code technology, one further problem is that much of the original content won't be displayable in the cell phone, since it will be in the form of big HTML pages. Alternatively, a user may be preoccupied and may simple wish to store a linkage code for subsequent retrieval. Therefore, it would be desirable to use the cell phone as a linkage code access device, without retrieving the associated content, but just for acquiring the linkage codes and uploading them to a code list server for later access by the user at a PC running the appropriate software.

DISCLOSURE OF THE INVENTION

In a first major aspect of the invention, provided is a method of and system for accessing a primary content file with a client device that is browser-based; i.e. that does not require a plug-in type program to be executed on the client device in addition to the browser. The user inputs into the client device a linkage code comprising a routing identification code and an item identification code, and the client device transmits to a URL-assembly server a data stream comprising the linkage code. The URL-assembly server

extracts the routing identification code from the data stream and then obtains a URL template associated with the routing identification code, the URL template comprising the name of a resolution server and at least one parameter field to be completed by the URL-assembly server. The URL-assembly server completes the URL template by filling in at least the item identification code and then sends the completed URL template to the resolution server named therein as a primary content URL request. The resolution server determines the location of the primary content file based on the item identification code and then provides the client device with the primary content file.

The URL-assembly server first obtains the URL template from a routing server. The URL-assembly server may then cache the URL template along with an expiration date for the URL template, which may be obtained from the routing server.

The data stream transmitted from the client device to the URL-assembly server may also further include a URL template selection code. In this event, then the URL-assembly server also extracts the URL template selection code from the data stream, and the URL template obtained by the URL-assembly server is also associated with the URL template selection code.

The URL template may be further completed by filling in least one parameter field with a device identification code, which may be included in the data stream transmitted from the client device to the URL-assembly server. The URL template may also be further completed by filling in least one parameter field with user data, which for example may be retrieved from a user database (populated by a user during a first registration process) located on a registration server.

The resolution server may directly return the primary content file, or it may redirect the client device to the primary content file by transmitting to the browser on the client device a primary URL for the primary content file. Alternatively, the primary URL may be sent to the client device via a proxy server, wherein the primary URL comprises an auto-request code that automatically redirects the client device to a content server containing the primary content file.

The linkage code input into the client device may for example be a bar code symbol that is scanned with a bar code scanning device coupled to the client device. The linkage code may also be a human-readable alphanumeric text string that may be typed in with a keypad connected to the client device.

The client device may be a wireless device such as a web-enabled cell phone, in which case a proxy server may used by the device for communication with the URL-assembly server, the resolution server and the content server.

Thus, in the method and system of the first major aspect of the invention, an existing browser-based client device such as a web-enabled cell phone or an internet kiosk, may be used to obtain content files from servers on the internet wherein the URL assembly steps are carried out on a server platform on the internet, instead of being done by a plug-in type module running with the client's browser.

In a second major aspect of the invention, provided is a method of and system for providing a primary content file to a client device wherein the primary content file will vary based on the display (or other) parameters of the client device, which is referred to as device-based routing. A linkage code comprising a routing identification code and

an item identification code is input into the client device, and the routing identification code and a client device identification code are then transmitted to a routing server. A URL template is obtained from the routing server, the URL template being associated with the routing identification code and the client device identification code, wherein the URL template includes the name of a resolution server and at least one parameter field to be completed by the client device. The URL template is completed by filling in the item identification code, such that the completed URL points to content suitable for display on the client device. The completed URL template is sent to the resolution server named therein to determine the location of the primary content file based on the item identification code and the client device identification code. A primary content URL that specifies the location of the primary content file is then sent to the client device, and the client device is redirected to a primary content server specified by the primary content URL.

For example, the client device may be a wireless device supporting WML content or HTML content, or it may be a personal computer supporting HTML content.

In a third major aspect of the invention, provided is a method of and system for providing a primary content file to a client device from a wireless device by using a stored code list service. In this case, the wireless device is used only to collect and transmit linkage codes to a list server (the linkage codes can be later accessed by any client device on the internet), and not to actually retrieve and display content associated with the linkage code. First, a linkage code is entered into a wireless transmitting device, wherein the linkage code comprises a routing identification code and an item identification code. The linkage code is then transmitted in a data stream to a list

server for storage therein, wherein the data stream includes a subscriber identification number associated with the wireless device. Later, a client device may be used to log into the list server, wherein the client device provides the subscriber identification number of the linkage code stored on the list server. The linkage code may then be retrieved from the list server and used to obtain a primary content file in accordance with one or more of various methods described herein.

The client device may be one of a group of devices comprising a personal computer, a web-based television, and a video console.

BRIEF DESCRIPTION OF THE DRAWING

FIG. 1 depicts a schematic overview of the wireless application protocol architecture.

FIG. 2 depicts a block diagram of an exemplary system of the present invention for a device directly connected to the Internet.

FIG. 2A depicts a block diagram of an alternative system of the present invention for a device connected to the Internet via the wireless application protocol.

FIG. 3A depicts a flow chart of how a linkage code is mapped to a content server.

FIG. 3B is a continuation of the flowchart of FIG. 3A.

FIG. 3C depicts a flow chart of how a new user is registered by the system of the invention.

FIG. 4 depicts an exemplary system of the present invention for the storage of linkage codes on list servers.

BEST MODE FOR CARRYING OUT THE INVENTION

The system of the present invention is a modification of the invention described in "SYSTEM AND METHOD OF USING MACHINE-READABLE OR HUMAN-READABLE LINKAGE CODES FOR ACCESSING NETWORKED DATA RESOURCES", copending U.S. Patent Application No. 09/543,178, filed April 5, 2000 by Hunter et al., previously incorporated herein by reference ("the copending application"). In the system described therein, a linkage code is a bar code that is scanned by a bar code scanner and input into a client software program that uses the decoded linkage code to request a URL template from an external server computer. The inputting of the code may also be performed by manually entering a text string associated with the code, such as by entering a UPC number found at the bottom of a typical UPC barcode. The linkage codes of the invention are not limited to UPC codes, however, and the invention supports European EAN codes, ISBN codes for books, as well as custom linkage code formats.

In a preferred embodiment of that invention, the linkage code includes two subcodes: a routing identification code ("RID") and an item identification code ("IID"). In the embodiment wherein the linkage code is a UPC code, the RID can be the manufacturer's portion of the UPC, whereas the IID can be the item code portion of the UPC. The client passes the RID to a routing server to obtain a URL link to a resolution server for that code, and the client completes the URL link by filing in the IID. The client then passes the completed URL link to the resolution server to obtain a target URL of content associated with the IID on the content server associated with the RID.

The two step resolution process allows for multiple resolution servers, thus providing scalability, with each server having its own database of target URLs. Since the

address of a resolution server for a particular RID changes infrequently with respect to number of times a user seeks to access the content server associated with the RID, the RID obtained from the routing server can be cached on the client for rapid lookup. The RID thus obtained can be associated with an expiration date so that the RID is periodically refreshed.

The system of the invention also includes a user database maintained by a registration server. The first time a user utilizes the invention to, for example, scan a UPC barcode to access a web site associated with the product, the user is directed to a registration procedure wherein the user is prompted to enter demographic data about him or herself. This data can include the user's name, address, age, gender, preferred language, and preferred interests. The registration server returns a user identification code ("UID") to the client, which caches it. The UID is passed to the routing server, which can then access the user data base and fill user data into the template URL. The template URL is returned to the client, which fills in the UID and IID to complete the lookup-URL. The client then passes the lookup-URL to the resolution server, which uses the user data along with a rules database to return a target-URL that addresses content specifically for that user. This feature of the invention is referred to as profiled routing.

Thus, the use of linkage codes is a powerful way of utilizing a general purpose computer to automatically access a web resource without having to type in a lengthy URL. Linkage codes are particularly useful in the context of wireless, hand-held web enabled devices such as cell phones, PDAs, or pocket personal computers ("pocket PCs"). Cell phones, for example, do not support the full alphabetic keyboard of a personal computer, and thus entering a full

URL for a web site is quite tedious. Most phones use a metaphor in which numeric buttons are pressed multiple times to scroll through several letters and/or punctuation marks, with either a button press or a pause indicating acceptance of the current letter. For example, www.amazon.com is entered on a cell phone numeric keypad as 99900262999966666002226666. On the other hand, the associated linkage code is merely 92801726. The all digit linkage code is shorter and easier to enter than the full URL, and much more intuitive to use. The advantage of linkage codes is even more apparent for those handheld devices that include barcode scanners, such as PDA's.

Although some wireless, hand-held web enabled devices, such as Palm Pilots or other PDAs, could easily be provided with the client plug-in required to map the linkage code into a URL, cell phones are not so easily adapted. There is also a large number of cell phones already in use. The inventors have thus found that it is preferable to locate this functionality on another server, referred to herein as a URL-assembly server. This enables any wireless device user to utilize linkage codes to access web content by merely accessing the appropriate page of the URL-assembly server that provides the mapping, without the necessity of installing the plug-in on the wireless client device.

Referring now to FIG. 2, an exemplary system configuration for a web-enabled device, such as a PDA that supports an HTML display, is depicted. Device 200 can execute a web browser whose interface is displayed in display area 210. When executing, the web browser can display the linkage code entry window, referred to as a go-window. The go-window includes a field 211 for entering a linkage code, and a button 212. A user can key in or write in a linkage code in field 211 and activate button 212 to find the associated web page. Alternatively, if device 200

supports a scanner 213, a barcode can be scanned in.

If device 200 is an Internet enabled device, such as a Palm VII PDA, it can transmit the linkage code just entered over the Internet to a URL-assembly server 202. The device 200 can also optionally transmit a user identification ("UID") to the URL-assembly server 202. If the device 200 is a WAP enabled cell phone that displays WML content, the transmission to the URL-assembly server 202 is typically mediated by a proxy server 201, shown in FIG 2A, that converts the WAP transmission into an HTTP compliant transmission. The URL assembly server 202 in turn communicates over the Internet with a registration server 203, which maintains a database of user information 214, and a routing server 204, which maintains a resolution server database 215. The URL-assembly server utilizes the RID portion of the linkage code, along with the UID, if available, to assemble a lookup URL in a manner described below. The lookup URL addresses a resolution server 205 that contains the target URL of the Internet content associated with the linkage code. The target URL received from the resolution server 205 redirects device 200 to the content server 206 containing the content associated with the linkage code.

The process by which linkage codes are mapped to content that is then downloaded is depicted in FIGS. 3A and 3B. A user begins the process by entering a linkage code in field 211 at step 300 in FIG 3A. The linkage code is transmitted to the URL-assembly server 202 at step 301. If a user is using the linkage code system for the first time, she would have to key in to the device 200 the name of the go-window in the traditional manner, by keying in the full name of the window, for example, www.paperclick.com. Once downloaded, however, the go-window can be bookmarked for easy subsequent retrieval.

The process by which a first-time user registers with the system is depicted in FIG. 3C. If the user is using a device that can transmit a unique device identifier, such as a PDA or cell phone using OpenWave's UP.Link proxy, she will be prompted at step 352 to register with the linkage code service. The user will be connected by the URL-assembly server 202 to the registration server 203. The registration server 203 can prompt the user at step 353 to enter various items of personal information, such as her name, address, age, gender, preferred language, and preferred interests. This information is stored at step 354 in user database 214. The user is assigned a UID so that she can be identified by the system. As part of this process, an entry can be made in the user database linking the UID to the unique device identifier. A given UID can even be linked multiple device identifiers.

Even if a client device does not support the transmission of a unique device identifier, it can still be identified to the system if the device's mini-browser supports standard authentication means, such as the storage of cookies. For this type of client, the registration server sends a cookie to the client's browser, which stores the cookie on the client device. Subsequent transmissions for the client to then URL-assembly server would then include the cookie.

The user data enables the linkage code system of the present invention to support the profiled routing feature of the client-based linkage system of the copending application. If, however, the device 200 does not support the transmission of a UID, the user of device 200 will remain anonymous to the linkage code system, and profiled routing is not available.

Continuing with FIG. 3A, once a linkage code has been received by the URL-assembly server 202, it is broken up at step 302 into its constituent parts, namely, the RID and the IID. The RID is passed to the routing server 204 at step 303 to obtain a URL template containing the address of the resolution server 205 associated with the IID. The resolution server 205 can actually be a front for any manufacturer's or vendor's server that can map a product code to a URL. This introduces the possibility that a given manufacturer or vendor could have a server for fielding WML queries that is different from servers fielding HTML queries. Since queries coming from the proxy server 201 typically indicate in the HTML header that the device 200 supports a WML browser, the URL-assembly server 202 can optionally include a URL template selection parameter in the data stream sent to the routing server so that a WML oriented template is returned by the routing server 204 to the URL-assembly server 202. The template selection parameter can also be used to ensure that the WML content ultimately returned to the user is not encapsulated in a frameset, as framesets are not supported by WML.

If the device 200 has been previously registered with the system, its UID will be included in the transmission to the URL-assembly server 202. In this situation, the URL-assembly server at step 304 passes the UID to the registration server which in turn uses the UID to retrieve user data for that user from the user database 214, and returns that data to the URL-assembly server 202.

The URL-assembly server 202 completes the URL-template at step 305. The URL template returned by the routing server 204 has at least one field for the URL-assembly server 202 to fill in. A typical URL template will look something like:

http://resolve.paperclick.com:8080/all/cmd?CMD=GET&TYPE=^TYP

E^&RID=^RID^&IID=^IID^&CODE=^CODE^, wherein the fields delimited by carets ("^") are to be filled in by the URL-assembly server. In the example shown, the fields to be filled in are the code type, the RID, the IID, and the full linkage code. If the linkage code is a UPC code equal to 051111128817, the RID would be 051111, the IID would be 12881, and the completed URL would be http://resolve.paperclick.com:8080/all/cmd?CMD=GET&TYPE=UPC&RID=051111&IID=12881&CODE=051111128817. There can also be fields for the UID, user data retrieved from the user database maintained by the registration server, and the template selection parameter. This list of fill-in codes is illustrative, and more fill-in fields can be supported and be within the scope of the invention.

The URL template also includes a field for a parameter indicative of the display device, i.e. what markup language the display device supports. This parameter could be the template selection parameter, or it could be a separate parameter. This enables the resolution server to list the addresses of multiple versions of a given page, a feature referred to a device-based routing. Thus, publishers can host web content on multiple formats accessible with different URLs, but use the same linkage code to access the content. Users are dynamically routed to the proper content based on the characteristics of the retrieving device.

The completed URL, referred to as a lookup-URL, is a reference to both a particular resolution server and an entry in that resolution server. The resolution server can be a front for any server, such as a vendor's server, that can map the IID to appropriate content on the Internet. The lookup-URL is returned at step 306 to the device 200, or the proxy server 201 if device 200 is a WAP device, and redirects the device 200, or the proxy server 201, to the resolution server. Continuing onto FIG. 3B, the resolution

server 205, at step 307, finds an appropriate target URL based on the information contained in the lookup-URL: the RID, the IID, and the user data if the user is registered. This ensures that the content customized to the user is subsequently returned. The resolution server includes lookup-tables and rules that ensure that a target URL to a content server 206 containing a web page in the correct display language is returned to the sender. The use of rules and tables to map the lookup-URL to a target-URL for content appropriate for a particular user is described in the copending application, and need not be repeated herein. The skilled artisan can easily extend the rules disclosed therein to support the device based routing feature of the present invention.

The target-URL returned by the resolution server normally redirects the sending device, either device 200 or proxy server 201, to the content server 206. In the case of a WAP compliant cell phone, however, having the proxy server perform the redirection means that the cell phone's mini-browser will not know of the redirection. Thus, when the cell phone device 200 receives the content, it would think the content had come from the server specified by the lookup URL, i.e. the resolution server 205, not the content server 206 specified by the target URL. If the returned content includes a relative URL or image reference, the device 200 will issue a request to the resolution server 205, not the content server 206. Therefore, the redirection to the content server 206 is not performed by the proxy server 201. Instead, at step 311, a data stream is returned to the WAP device 200 that includes the target-URL hyperlink along with an auto-click code to force the device 200 at step 312 to automatically make the request to the content server 206. If the device is directly connected to the Internet, that device is redirected at step 309 to the content server 206. Finally, the content is downloaded to the device 200 at step

310.

In addition to the functionality described above, the system of the present invention supports the demographic reporting, obfuscation/de-obfuscation of the UID, and profiled routing features disclosed in the copending application. In addition, the device based routing feature of the present invention can also be included with the invention of the copending application. The linkage client software disclosed in the copending application can easily be modified to include a device indicator field in the data stream transmitted to the routing server. This enables the routing server to select a URL template appropriate for the display device.

One application of profiled routing is the ability to streamline registration for cell-based services. Where there are user-specific parameters such as presentation language, a user registering for a service via a linkage code could have profile information passed from the user database 214 into the service registration process. This would potentially allow the registration form to be pre-populated with the user's information, thus allowing the user to simply confirm the information, rather than having to laboriously enter it.

In many cases, a cell phone user, because of the sparse nature of the WML or HDML display as compared with an HTML form, or because the user is preoccupied with another task, may not want to actually download content directly to her cell phone upon entering a linkage code, but may wish to save the linkage code for subsequent retrieval by a personal computer that supports HTML. In another aspect of the invention, a registered user can store linkage codes on a list server for subsequent retrieval. Referring now to Figure 4, by using a simple WML or HDML form 401 on the cell

phone 400, a linkage code is entered into field 408 in the form 401 via the keypad 403, and then submitted via a store button 409 over the wireless network 404 and Internet 405 to a list server 406, where it would be stored. The wireless transactions pass on the user's UID, so the system can use this to indicate which list the code should be added to. Later, when the user sits down at his or her PC 407, he or she can log into the server 406, and download the accumulated stored codes to the PC 407, just as if they had used the linkage client software. A user does not even need a PC - he or she could retrieve the codes via a WebTV, video game console (the newest generation of video game consoles apparently have browsers and modems built in) or any other device that has an embedded browser. The user's login/password can be used as the authentication means, and can be tied to their UID via the registration process.

By using the stored linkage code list service described herein, a user can download stored linkage codes lists to a client device anywhere in the world. The user can even upload linkage codes from a desktop client to the list server, then download them onto any other client device, such as a laptop computer via the web.

The system of the present invention is currently implemented on a Windows NT platform, although the system can be adapted to operate on other operating systems, such as Unix or Linux, or the Macintosh. The registration, routing and resolution servers are currently implemented as stand-alone programs, written in C++, that run as services under Windows NT, and the URL-assembly server is currently implemented as a component of the routing server. Other implementation are, of course possible, and the servers could be implemented as ISAPI DLLs running on a web server that communicate with a Microsoft SQL database server via ODBC, as CGI programs or as Java servlets. The routing

server can also be implemented as a stand-alone program. Although these components have been described as if they are physically distinct machines, the skilled artisan will understand that they can be distinct processes running on the same machine.

The system of the invention is not limited to the embodiment disclosed herein. It will be immediately apparent to those skilled in the art that variations and modifications to the disclosed embodiment are possible without departing from the spirit and scope of the present invention. The invention is defined by the appended claims.

WHAT IS CLAIMED IS:

1. A method of accessing a primary content file with a client device comprising the steps of:

(a) inputting into the client device a linkage code comprising a routing identification code and an item identification code;

(b) transmitting from the client device to a URL-assembly server a data stream comprising the linkage code;

(c) extracting by the URL-assembly server the routing identification code from the data stream;

(d) obtaining by the URL-assembly server a URL template associated with the routing identification code, the URL template comprising the name of a resolution server and at least one parameter field to be completed by the URL-assembly server;

(e) completing at the URL-assembly server the URL template by filling in the at least one parameter field;

(f) sending the completed URL template to the resolution server named therein as a primary content URL request;

(g) determining at the resolution server the location of the primary content file based on the item identification code; and

(h) the resolution server providing the client device with the primary content file.

2. The method of claim 1, wherein the URL template is obtained from a routing server.

3. The method of claim 2, further comprising the step of caching the URL template on the URL-assembly server, along with an expiration date for the URL template.

4. The method of claim 3, wherein the expiration date for the URL template is obtained from the routing server.

5. The method of claim 3, further comprising the step of retrieving the URL template from the routing server when the current date is later than the expiration date.

6. The method of claim 1, wherein the data stream transmitted from the client device to the URL-assembly server further comprises a URL template selection code.

7. The method of claim 6, further comprising the step of the URL-assembly server extracting the URL template selection' code from the data stream.

8. The method of claim 7, wherein the URL template obtained by the URL-assembly server is also associated with the URL template selection code.

9. The method of claim 1, wherein the at least one parameter field is filled in with the item identification code by the URL-assembly server

10. The method of claim 1, wherein the URL template is further completed by filling in the least one parameter field with a device identification code.

11. The method of claim 10, wherein the device identification code is included in the data stream transmitted from the client device to the URL-assembly server.

12. The method of claim 1, wherein the URL template is further completed by filling in least one parameter field with user data.

13. The method of claim 12, wherein the user data is retrieved from a user database located on a registration server.

14. The method of claim 13, wherein the user database is populated by a user during a first registration process.

15. The method of claim 13, further comprising a second registration process wherein the user uses the linkage code to register for a service, and the second registration process uses user data retrieved from said user database.

16. The method of claim 1, wherein the resolution server provides the client device with the primary content file by transmitting to the client device a primary URL for the primary content file, the primary URL comprising an auto-request code that automatically redirects the client device to a content server containing the primary content file.

17. The method of claim 16, wherein the primary URL is sent to the client device via a browser.

18. The method of claim 16, wherein the primary URL is sent to the client device via a proxy server.

19. The method of claim 1, wherein the linkage code is a bar code symbol, and wherein the step of inputting comprises the step of scanning the bar code symbol with a bar code scanning device connected to the client device.

20. The method of claim 1, wherein the linkage code is a human-readable alphanumeric text string, and wherein the step of inputting comprises the step of typing in the alphanumeric text string with a keypad connected to the client device.

21. The method of claim 1, wherein the client device is a wireless device.

22. The method of claim 21, further comprising a proxy server, by means of which the wireless device communicates with the URL-assembly server, the resolution server and the content server.

23. A computer system for accessing a primary content file on a primary content server over a computer network with a client device, comprising:

(a) a client device interconnected to the computer network;

(b) a URL-assembly server interconnected to the computer network;

(c) a resolution server interconnected to the computer network; wherein

the client device comprises:

means for inputting a linkage code comprising a routing identification code and an item identification code;

means for transmitting a data stream comprising the linkage code to the URL-assembly server;

the URL-assembly server comprises:

means for extracting the routing identification code from the data stream received from the client device;

means for obtaining a URL template associated with the routing identification code, the URL template comprising the name of a resolution server and at least one parameter field to be completed by the URL-assembly server;

means for completing the URL template by filling in the at least one parameter field;

means for sending the completed URL template to the resolution server named therein as a primary content URL request; and

the resolution server comprises:

means for determining the location of the primary

content file based on the item identification code; and

means for providing the client device with the primary content file.

24. The computer system of claim 23, further comprising a routing server from which the URL-assembly server obtains the URL template.

25. The computer system of claim 24, wherein the URL-assembly server caches the URL template, along with an expiration date for the URL template.

26. The computer system of claim 25, wherein the expiration date for the URL template is obtained from the routing server.

27. The computer system of claim 25, wherein the URL-assembly server further comprises means for retrieving the URL template from the routing server when the current date is later than the expiration date.

28. The computer system of claim 23, wherein the data stream transmitted from the client device to the URL-assembly server further comprises a URL template selection code.

29. The computer system of claim 28, wherein the URL-assembly server further comprises means for extracting the URL template selection code from the data stream.

30. The computer system of claim 29, wherein the URL template obtained by the URL-assembly server is also associated with the URL template selection code.

31. The computer system of claim 23, wherein the URL-assembly server further comprises means for completing the

URL template by filling in least one parameter field with the item identification code.

32. The computer system of claim 23, wherein the URL-assembly server further comprises means for completing the URL template by filling in least one parameter field with a device identification code.

33. The computer system of claim 32, wherein the client device further comprises means for including the device identification code in the data stream transmitted from the client device to the URL-assembly server.

34. The computer system of claim 23, wherein the URL-assembly server further comprises means for further completing the URL template by filling in least one parameter field with user data.

35. The computer system of claim 34, further comprising a registration server comprising a user database, wherein the user data is retrieved from the user database.

36. The computer system of claim 35, wherein the user database is populated by a user during a first registration process.

37. The computer system of claim 35, further comprising a second registration process wherein the user uses the linkage code to register for a service, and the second registration process uses user data retrieved from said user database.

38. The computer system of claim 23, wherein the resolution server comprises means for providing the client device with the primary content file by transmitting to the client device a primary URL for the primary content file,

the primary URL comprising an auto-request code that automatically redirects the client device to a content server containing the primary content file.

39. The computer system of claim 38, wherein the primary URL is sent to the client device via a browser.

40. The computer system of claim 38, wherein the primary URL is sent to the client device via a proxy server.

41. The computer system of claim 23, wherein the means for inputting a linkage code comprises a bar code scanning device for scanning a linkage code in the form of a bar code symbol.

42. The computer system of claim 23, wherein the means for inputting a linkage code comprises a keypad for entering a linkage code in the form of a human-readable alphanumeric text string.

43. The computer system of claim 23, wherein the client device is a wireless device.

44. The computer system of claim 43, further comprising a proxy server, by means of which the wireless device communicates with the URL-assembly server, the resolution server and the content server.

45. A method of providing a primary content file to a client device comprising the steps of:

(a) inputting into the client device a linkage code comprising a routing identification code and an item identification code;

(b) transmitting to a routing server the routing identification code and a client device identification code, and obtaining from the routing server a URL template

associated with the routing identification code and the client device identification code, the URL template comprising the name of a resolution server and at least one parameter field to be completed by the client device;

(c) completing the URL template by filling in the item identification code, the completed URL pointing to content suitable for display on the client device;

(d) sending the completed URL template to the resolution server named therein to determine the location of the primary content file based on the item identification code and the client device identification code; and

(e) sending a primary content URL that specifies the location of the primary content file to the client device and redirecting the client device to a primary content server specified by the primary content URL.

46. The method of claim 45, further comprising the step of providing the primary content file to the client device from the primary content server.

47. The method of claim 45, wherein the client device is a wireless device supporting WML content.

48. The method of claim 45, wherein the client device is a wireless device supporting HTML content.

49. The method of claim 45, wherein the client device is a wireless device supporting HDML content.

50. The method of claim 45, wherein the client device is a personal computer supporting HTML content.

51. A system for providing a primary content file to a client device over a computer network, comprising:

(a) a client device interconnected to the computer network;

(b) means for inputting into the client device a linkage code comprising a routing identification code and an item identification code;

(c) means for transmitting to a routing server the routing identification code and a client device identification code, and means for obtaining from the routing server a URL template associated with the routing identification code and the client device identification code, the URL template comprising the name of a resolution server and at least one parameter field to be completed by the client device;

(d) means for completing the URL template by filling in the item identification code, the completed URL pointing to content suitable for display on said client device;

(e) means for sending the completed URL template to the resolution server named therein to determine the location of the primary content file based on the item identification code and the client device identification code; and

(f) means for sending a primary content URL that specifies the location of the primary content file to the client device and redirecting the client device to a primary content server specified by the primary content URL.

52. The system of claim 51, further comprising means for providing the primary content file to the client device from the primary content server.

53. The system of claim 51, wherein the client device is a wireless device supporting WML content.

54. The system of claim 51, wherein the client device is a wireless device supporting HDML content.

55. The system of claim 51, wherein the client device is a wireless device supporting HTML content.

56. The system of claim 51, wherein the client device is a personal computer supporting HTML content.

57. A method of providing a primary content file to a client device from a wireless device, said method comprising the steps of:

(a) entering a linkage code into a wireless transmitting device, wherein the linkage code comprises a routing identification code and an item identification code;

(b) transmitting the linkage code in a data stream to a list server for storage therein, wherein said data stream includes a subscriber identification number associated with the wireless device;

(c) logging in to the list server from a client device, wherein the client device provides the subscriber identification number of the linkage code stored on the list server; and

(d) retrieving the linkage code from the list server.

58. The method of claim 57, wherein a user provides the client device with the subscriber identification number.

59. The method of claim 57, wherein the client device is one of a group of devices comprising a personal computer, a web-based television, and a video console.

60. The method of claim 57, further comprising the step of utilizing the linkage code to enable the client device to retrieve a primary content file associated with the item identification code from a content server associated with the routing identification code.

61. A system for providing a primary content file to a client device from a wireless device over a computer network, comprising:

(a) a wireless device interconnected to the computer network;

(b) a list server interconnected to the computer network, the list server comprising a storage means;

(c) a client device interconnected to the computer network; wherein

the wireless device comprises:

means for inputting a linkage code comprising a routing identification code and an item identification code;

means for transmitting the linkage code in a data stream to the list server for storage therein, wherein said data stream includes a subscriber identification number associated with the wireless device; and

the client device comprises:

means for logging into the list server and providing the subscriber identification number of the linkage code stored on the list server; and

means for retrieving the linkage code from the list server.

62. The system of claim 61, wherein the client device further comprises means for a user to provide the client device with the subscriber identification number.

63. The system of claim 61, wherein the client device is one of a group of devices comprising a personal computer, a web-based television, and a video console.

64. The system of claim 61, further comprising means for utilizing the linkage code to enable the client device to retrieve a primary content file associated with the item identification code from a content server associated with the routing identification code.

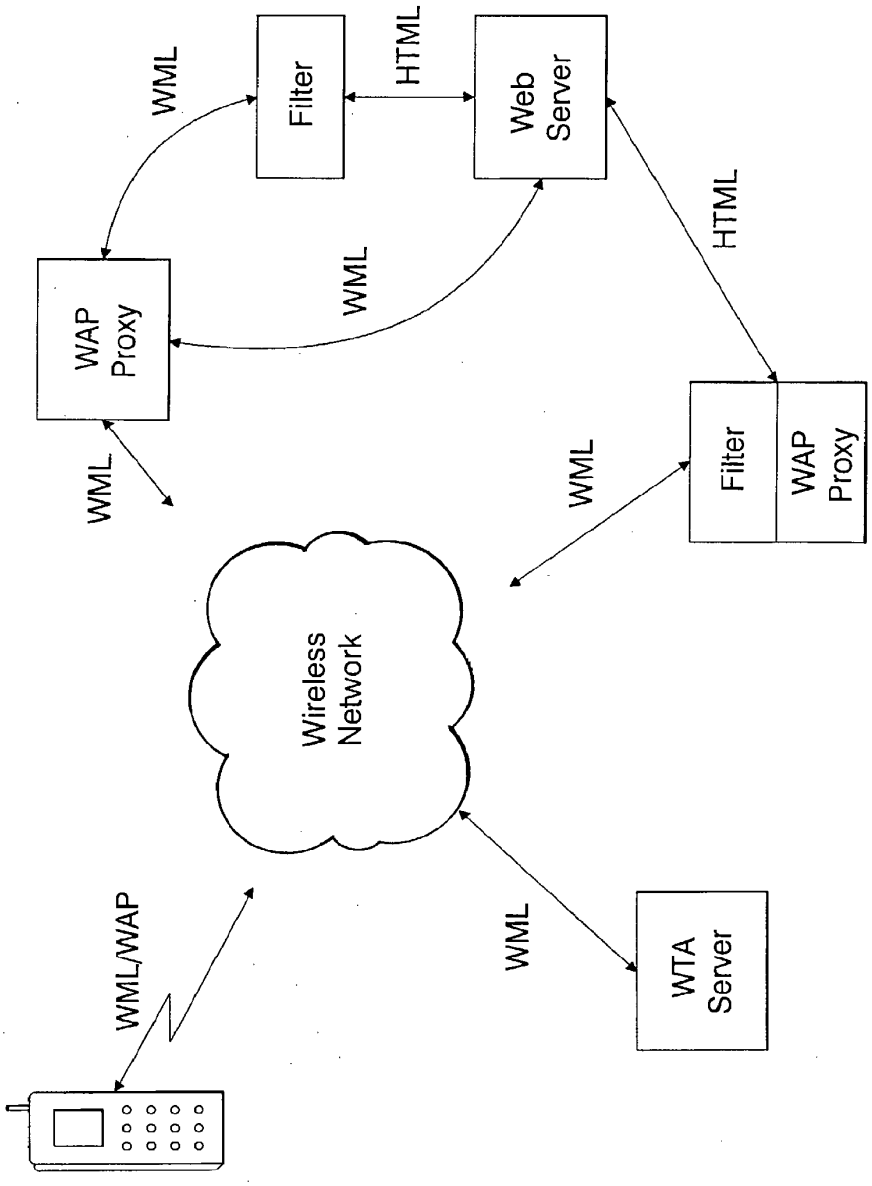

WML
Filter
HTML
Web Server
WAP Proxy
WML
HTML
WML
Filter
WAP Proxy
WML
Wireless Network
WML/WAP
WML
WTA Server

FIG. 1

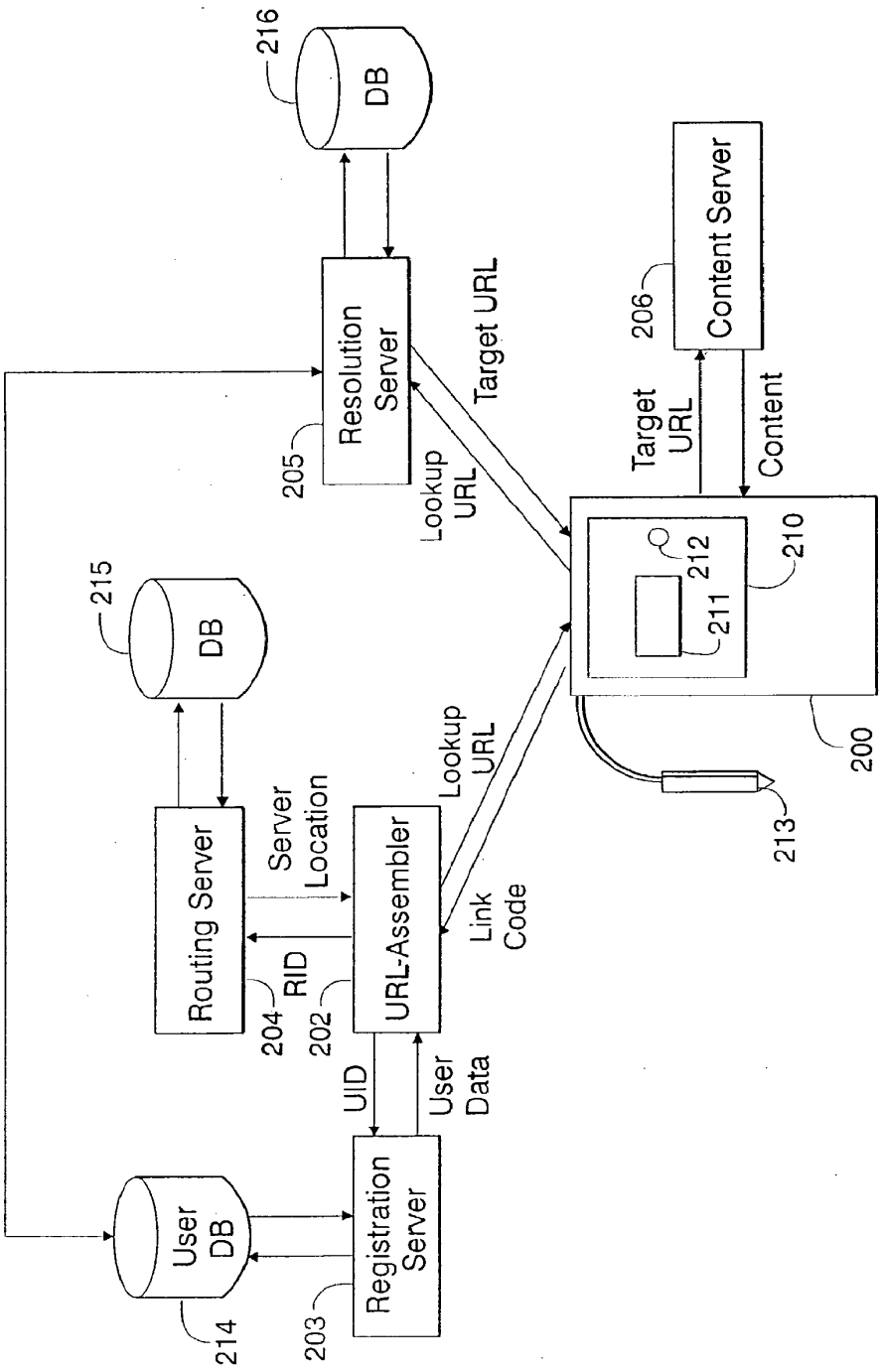

216
DB
Resolution Server
205
Target URL
Lookup URL
206
Content Server
Target URL
Content
215
DB
210
212
211
Routing Server
Server Location
RID
204
202
URL-Assembler
Lookup URL
Link Code
200
213
UID
User Data
User DB
214
Registration Server
203

FIG. 2

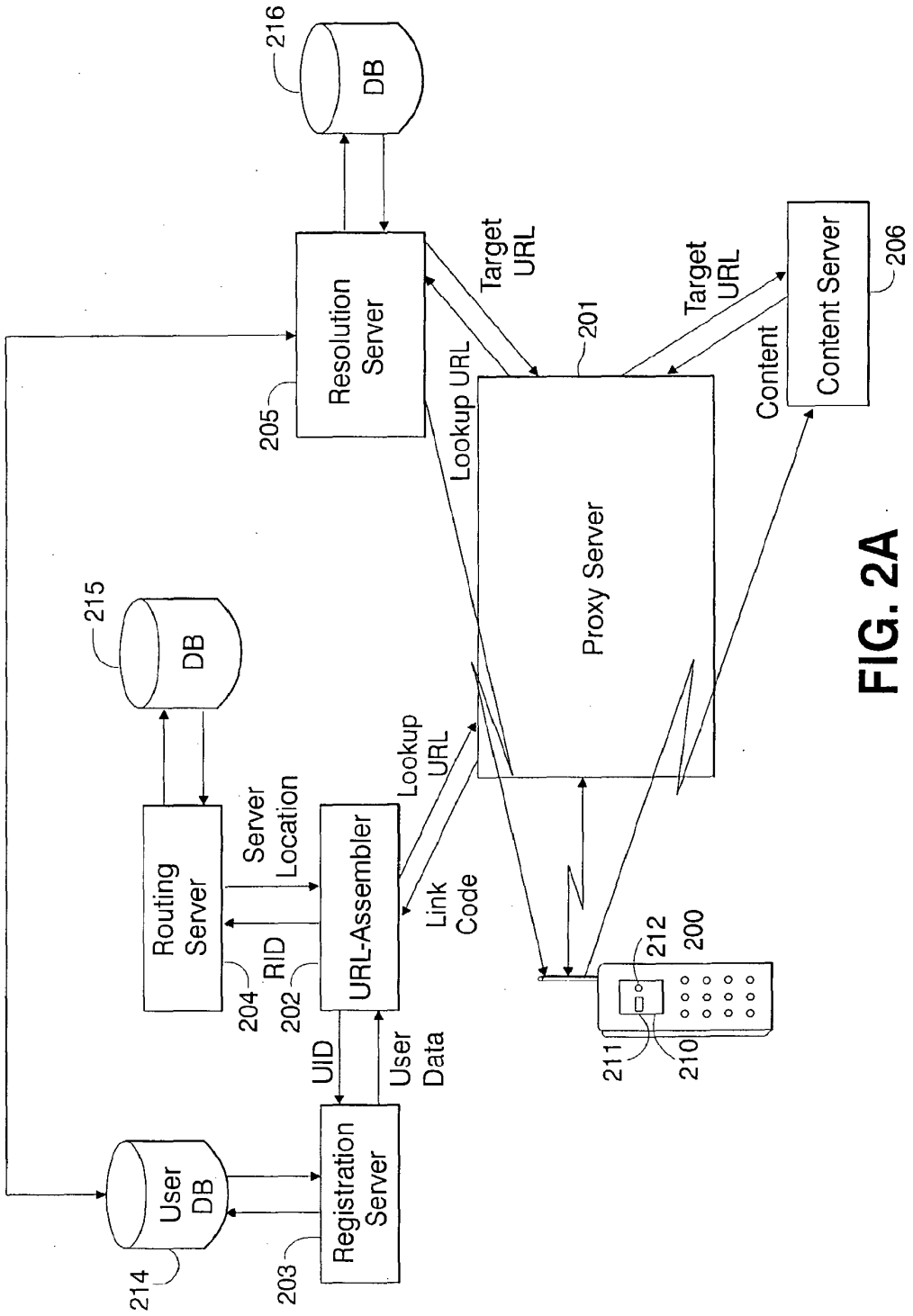

216
DB
Resolution Server
205
Target URL
Lookup URL
201
Target URL
Content
Content Server
206
Proxy Server
215
DB
Routing Server
204
Server Location
RID
URL-Assembler
202
Lookup URL
Link Code
212
200
211
210
UID
User Data
User DB
214
Registration Server
203

FIG. 2A

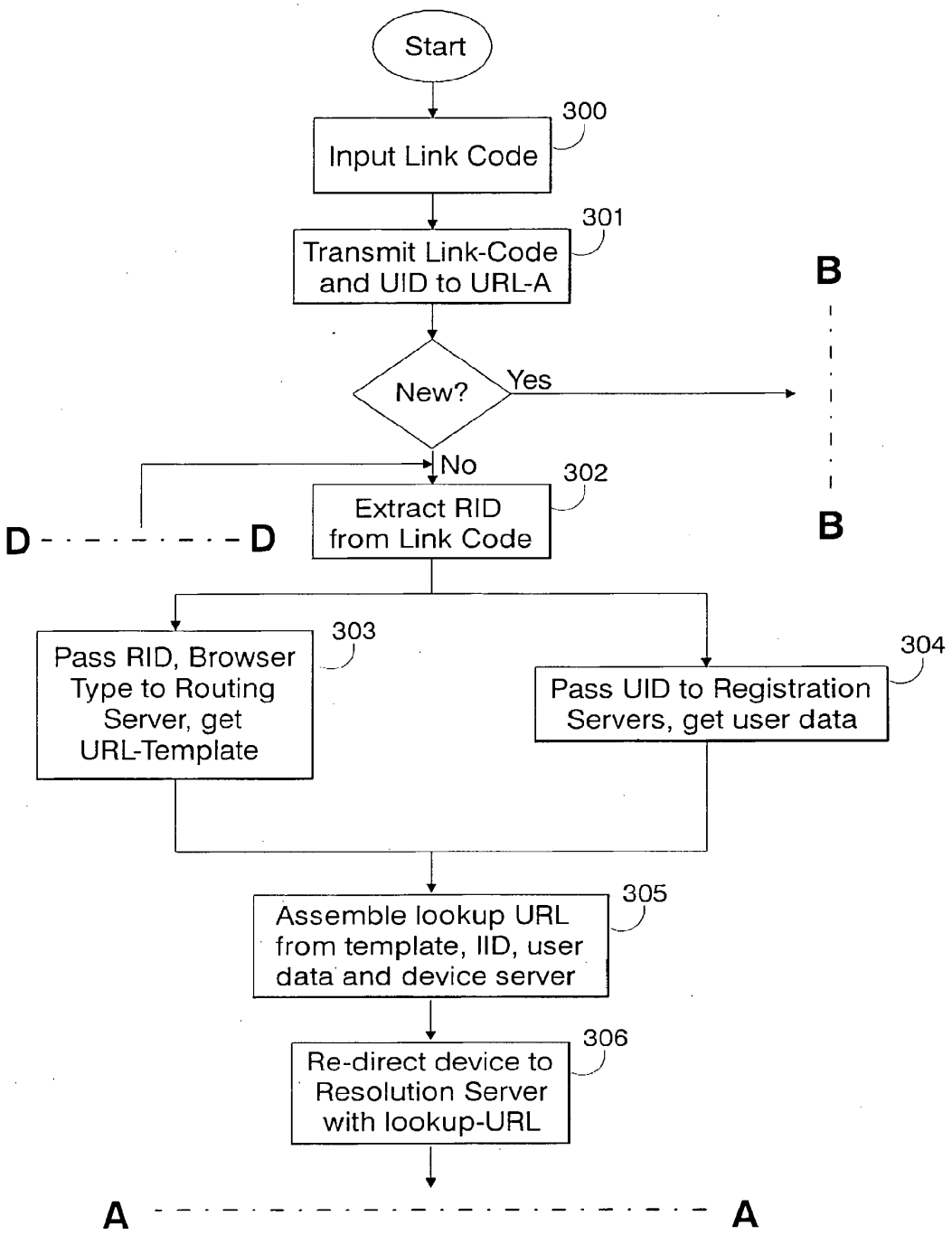

Start
300
Input Link Code
301
Transmit Link-Code and UID to URL-A
B
New?
Yes
No
302
Extract RID from Link Code
D
D
B
303
Pass RID, Browser Type to Routing Server, get URL-Template
304
Pass UID to Registration Servers, get user data
305
Assemble lookup URL from template, IID, user data and device server
306
Re-direct device to Resolution Server with lookup-URL
A
A

FIG. 3A

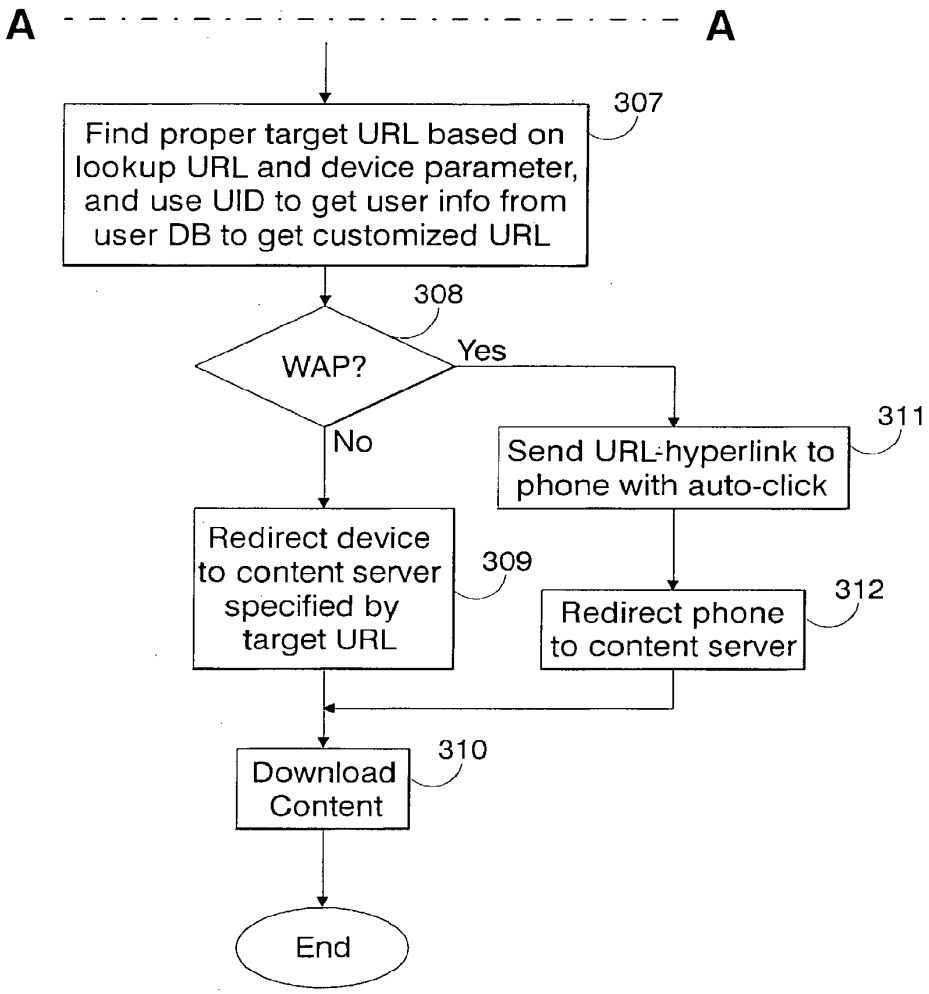

A
A
Find proper target URL based on lookup URL and device parameter, and use UID to get user info from user DB to get customized URL
307
308
WAP?
Yes
No
Send URL-hyperlink to phone with auto-click
311
Redirect device to content server specified by target URL
309
Redirect phone to content server
312
Download Content
310
End

FIG. 3B

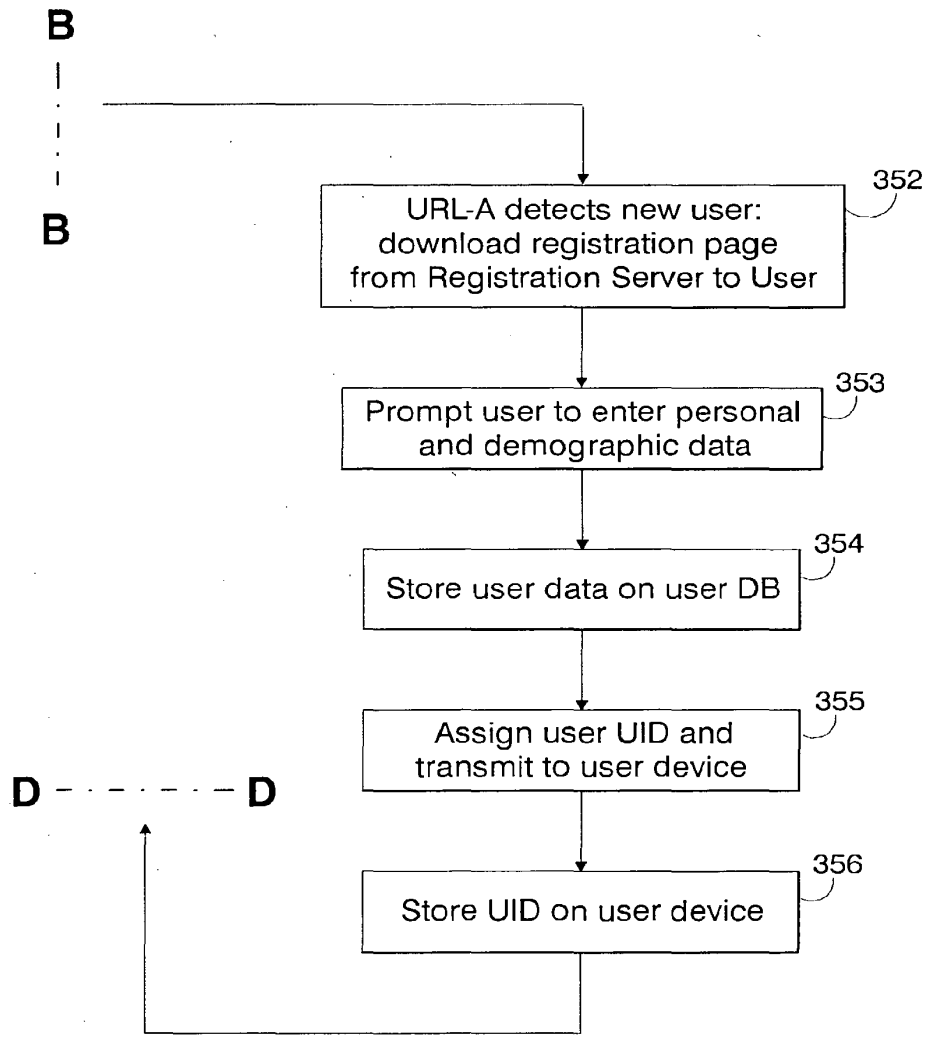

B
B
URL-A detects new user: download registration page from Registration Server to User
352
Prompt user to enter personal and demographic data
353
Store user data on user DB
354
Assign user UID and transmit to user device
355
D
D
Store UID on user device
356

FIG. 3C

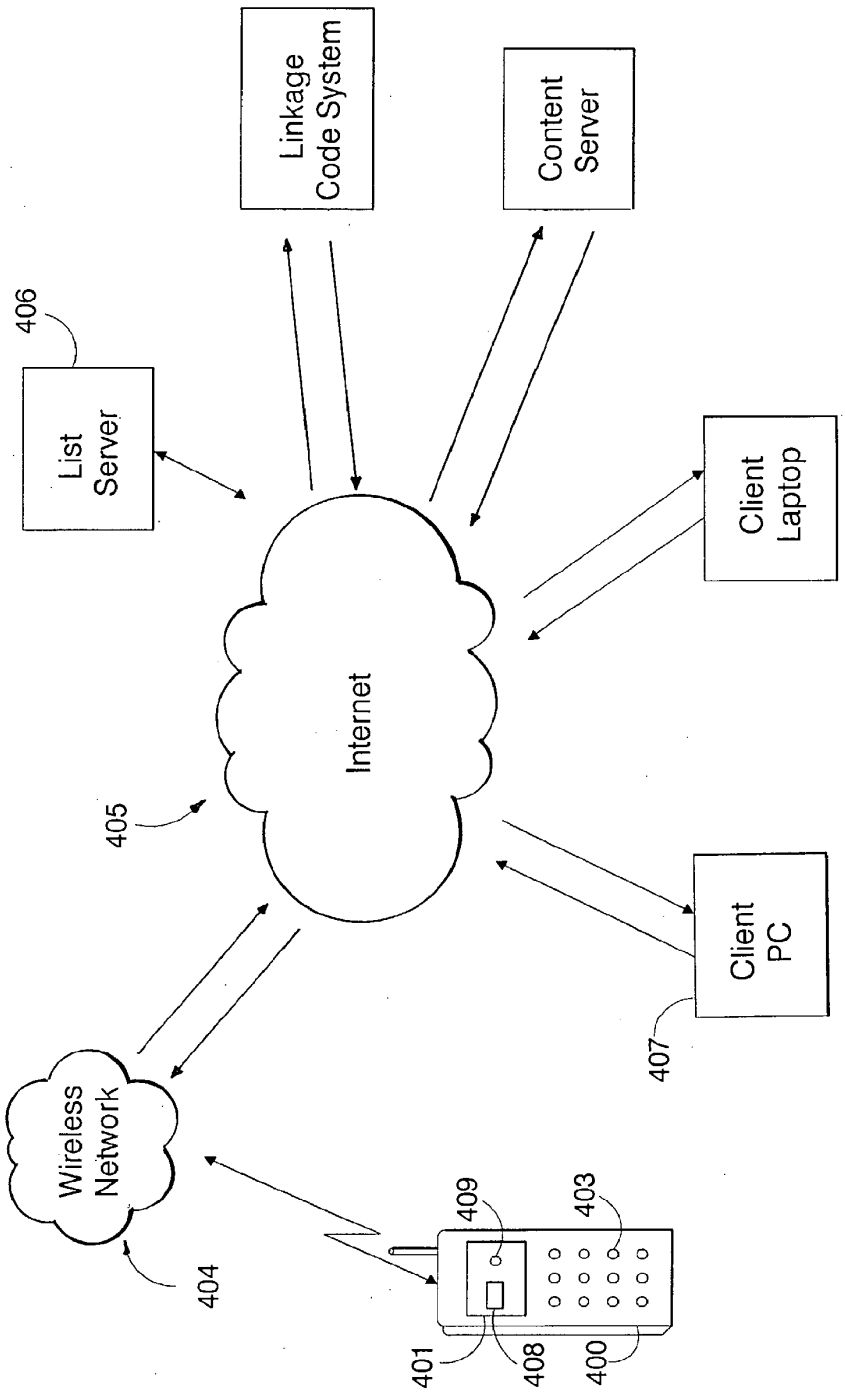

Linkage Code System
Content Server
406
List Server
Client Laptop
Internet
405
Client PC
407
Wireless Network
404
409
403
401
408
400

FIG. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/10336

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06F 15/16, 15/173
US CL :709/219, 229

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 709/219, 229

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WEST, STN, IEEE

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	US 5,804,803 A (CRAGUN et al.) 08 SEPTEMBER 1998, Fig. 1, 1B, an abstract, col. 2, lines 45-55, col. 5, lines 50-65, cols. 6-12	1, 23, 45, 51, 57, 61 --- 2-22, 24-44, 46-50, 52-56, 58-60, 62-64
Y	US 5,905,248 A (RUSSELL et al.) 18 MAY 1999, an abstract, Figs. 1 and 3, cols. 25-28.	1-64
Y	US 5,933,829 A (DURST et al.) 03 AUGUST 1999, an abstract, Figs. 1-3, cols. 10-14.	1-64
Y	US 5,986,651 A (REBER et al.) 16 NOVEMBER 1999, an abstract, Figs. 1-6, cols. 20-24.	1-64

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search	Date of mailing of the international search report
31 MAY 2001	28 JUN 2001
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer KRISNA LIM Telephone No. (703)-305-9672

Form PCT/ISA/210 (second sheet) (July 1998)★

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/10336

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A, P	US 6,064,979 A (PERKOWSKI) 16 MAY 2000, an abstract, Figs. 1, 1A, cols. 22-28.	1-64

Form PCT/ISA/210 (continuation of second sheet) (July 1998)*

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
20 December 2001 (20.12.2001)

PCT

(10) International Publication Number
WO 01/97128 A1

(51) International Patent Classification[7]: G06F 17/60

(21) International Application Number: PCT/KR01/00997

(22) International Filing Date: 11 June 2001 (11.06.2001)

(25) Filing Language: Korean

(26) Publication Language: English

(30) Priority Data:
2000/31945 10 June 2000 (10.06.2000) KR

(71) Applicant *(for all designated States except US)*: **MARKANY INC.** [KR/KR]; Ssanglim Bldg. 10Fl., 151-11, Ssanglim-dong, Chung-gu, Seoul 100-400 (KR).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **CHOI, Jong-Uk** [KR/KR]; 2-1301 Seongwon APT., 1, Wooi-dong, Dobong-gu, Seoul 142-090 (KR). **LEE, Won-Ha** [KR/KR]; 106-1704, Ssangryong APT., 64, Imun 3-dong, Dongdaemun-gu, Seoul 130-083 (KR).

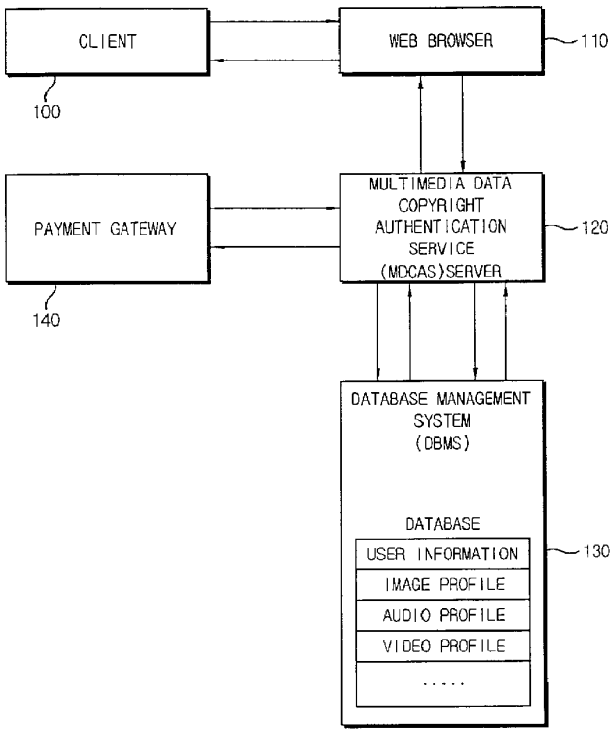
(74) Agent: **KOREANA PATENT FIRM**; Dong-Kyong Bldg. 824-19, Yoksam-dong, Kangnam-ku, Seoul 135-080 (KR).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: SYSTEM AND METHOD OF PROVIDING AND AUTHENTICATING WORKS OF AUTHORSHIP BASED ON WATERMARK TECHNIQUE



(57) Abstract: The supply of works of authorship and their authentication are provided by a service system and method for supplying and authenticating digital data works of authorship based on a watermarking system. The works of authorship including invisible information on a copyright is provided through this system. In the event that illegal use of original works of authorship is sensed, the illegal copy and use of multimedia contents could be prevented by presenting accurate technical evidence to support the information on a copyright through authentication of an authentication organization and thereby the protection of a copyright is possible. Authentication of a copyright of the contents is performed by the steps of imbedding and extracting of a digital watermark such as a logo or signature of a copyright holder on a data.

WO 01/97128 A1

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SYSTEM AND METHOD OF PROVIDING AND AUTHENTICATING WORKS OF AUTHORSHIP BASED ON WATERMARK TECHNIQUE

Technical Field

The present invention relates to a service system and method for supplying and authenticating a copyright based on a method and device, in which a watermark is embedded into the data of a digital content (digital image, digital audio, digital video, digital document, etc.) and the embedded watermark is extracted. More particularly, this invention relates to a service system and method for supplying and authenticating a copyright, which are capable of protecting a copyright of original data by selling or providing data, in which copyright information is hidden in an image watermark form, to users on the internet, and settling copyright-involved distribution by using an authentication result from a comparison of information extracted from the hidden copyright information with the copyright information previously registered in a certificate authority, thereby preventing illegal copying and distribution of the data.

Description of Prior Art

With the recent rapid spread of internet services, the population of internet users in Korea reached 14.8 millions as of the end of 1999. Under such circumstances of sharp growing Internet, presently, information such as text, still images and simple moving pictures is mainly in distribution. However, owing to the spread of various forms of video/audio transmission media, internet transmission of all sorts of multimedia data is on an increasing trend.

Meanwhile, it is believed that papers, magazines, audio, radio, TV programs,

and electronic libraries presently being operated free on the internet will be converted to chargeable services in the future. Domestic content providers are pushing forward chargeable services of contents centering on small payment service companies and PC telecommunication companies. According to such tendency, providers of digital information or content owners will need to assert and protect their copyrights on the information (digital copyrighted materials such as images, audio, video, etc.) they own, and request a reasonable compensation for use in the case of providing their copyrighted materials to others.

However, digital copyrighted materials have so many problems in their copyrights protection in view of their features. Since desired information could be easily stored or converted, multiple copies of information are possible without constraint in a free-accessible space like the internet. Further, even after copying, there can be maintained the same data information as that of the original, moreover, its distribution is possible without a copyright owner's consent, and thus copyright protection of a copyright owner is difficult.

As such, copyright infringements are increasing day by day, many legal, systemic and technical resolutions to solve these problems have been formulated. For this purpose, there are provided certain technical solutions, such as encryption of data by which an encrypted data is available to only a specific user and a digital watermarking technique where copyright information is hidden in data, which data is then transmitted to a user without the data being deformed outwardly.

Among these technical solutions, as a watermarking is a copyright protection technique of digital data, it can be considered a copyright indication or verification technique for a digital content. This technique makes it possible to detect counterfeit

or alteration of data and assert the ownership of a copyright holder by hiding copyright information in data and extracting again the hidden copyright information.

As watermarking embedding methods, there are provided a Spatial Method in which a subtle transformation is applied to data such as pixels of a screen and the like to be used as a watermark; Discrete Cosine Transform (DCT) in which digital-type data are converted to analog signals of frequency components and then a watermark converted by the same method is embedded; Fast Fourier Transform (FFT); and a Frequency Domain Method using wavelet transform.

A watermarking technique adopted by the present invention for copyright protection is to embed a "mark" which protects the original information. Such method is divided into a visible watermarking technique perceptible to the human eye and an invisible watermarking technique imperceptible to the human eye.

As the visible watermarking technique is a type of overwriting copyright information on the original, it generally prevents a user from illegally eliminating the copyright information. However, since the copyright information is visually displayed on the original, it has a disadvantage that it is difficult to preserve the value of the original. The visible watermarking technique is technically realizable without difficulty but it damages the content, thus making it difficult to be used in the electronic business.

In contrast, since the invisible watermarking technique is invisible to the human eye, it can protect a copyright without damaging the original. Also, in case that a copyright-related problem occurs, a copyright holder can claim his copyright based on the copyright information extracted from the original. Currently most watermarking techniques employ this invisible method. Accordingly, in case where the invisible

watermarking technique is used, even if images or audio/video files are distributed across the internet, users could not perceive at all whether a watermark is embedded in the copyrighted material distributed. Even a copyright holder who embedded a watermark could not find the difference between a file before the watermark is embedded and that after the watermark is embedded.

Conventional methods for embedding/extracting a watermark are directed to "a watermarking method based on detection" that embeds a user's ID (number, name) or simple data into a copyrighted material and detects them. In such an image watermarking method, presently, the existence of a watermark is determined by measuring a correlation between a watermark-embedded image and a user ID after embedding a user ID composed of 48 bits or so binary codes into image data. That is, if the correlation coefficient shows 0, the data is determined not to include the user ID, however, if the correlation coefficient shows values other than 0, the data is determined to include the user ID. Digimarc, BlueSpike, MediaSec and AlphaTech, etc., that are now providing a watermarking technique of the image field, generally use the above method.

However, under the present circumstances that embedding or detecting work of a specific ID as a watermark is performed directly by users, a big problem is how to establish a copyright on the relevant copyrighted material or settle such dispute in the event that copyright-related disputes arise.

Also, even in a case where copyrighted material is distributed through a legal course, there is a problem such that a copyright owner could not know the extent of the use and detection of such watermark is completed.

Furthermore, even if a watermark on a copyrighted material is embedded, since

such embedment is merely performed by either a user or a copyright holder but does not contain information on the copyrighted material according to its characteristics, it is impossible to find out a distribution channel of such copyrighted material.

Also, since the aforementioned methods use a method for embedding information that has no direct relationship with a copyright as binary code, there has a limitation that it is not easy to obtain information about a copyright from copyrighted material containing an embedded watermark.

Summary of the Invention

It is therefore an object of the present invention to overcome the above-described problems in prior art. In order to protect a copyright of digital multimedia data distributed on the internet, there is provided a service system and method for supplying and authenticating copyrighted materials by providing a service capable of verifying user information using a watermark embedding and extracting method based on extraction.

It is another object of the present invention to provide a system and method for protecting a copyright of a content provider (i.e., a copyright holder, and a seller to whom a duplication right of copyrighted material is assigned from the copyright holder) by embedding a watermark including user information into various digital content data and then distributing the content containing a copyright on the internet.

It is another object of the present invention to provide a means capable of verifying and authenticating an original by applying this technique to electronic documents exchangeable on the internet (various administrative civil affair documents, contracts, principle documents of companies, electronic publications, receipts,

purchasing documents, etc.) in order that parties can safely exchange electronic documents on the internet.

It is another object of the present invention to establish the just internet business culture by preventing the loss of information and the distribution of illegally copied data on the internet, which is possible by embedding main information into digital data in a watermark form and then transmitting it.

It is another object of the present invention to discover a distribution channel of the corresponding copyrighted material by transmitting copyright-related information which is embedded as a watermark after sorting the information on a copyright holder and users.

In order to achieve the above-mentioned objects, the service system for supplying copyrighted material according to the present invention includes a database for storing information on a copyrighted materials, and a copyrighted material supplying means for supplying the copyrighted material, at a user's request through the internet, into which copyright information is embedded as a watermark, said copyrighted material being received from said database.

Further, the service system for authenticating a copyrighted material according to the present invention includes a database for storing information on the copyrighted material, and a copyrighted material authenticating means for performing authentication on the copyrighted material supplied from a user, at the user's request through the internet, after comparing on extracted watermark embedded in the copyrighted material of said user with the information on said copyrighted material stored in said database.

The service method for supplying a copyrighted material according to the present invention includes a step of registering a copyrighted material supplied by a

copyright holder through the internet and copyright information of said copyrighted material; a step of embedding said copyright information into said copyrighted material as a watermark at a user's request for the supply of said registered copyrighted material; and a step of providing the watermark-embedded copyrighted material to said user.

The service method for authenticating a copyrighted material according to the present invention includes a step of receiving a copyrighted material at a user or copyright holder's request through the internet for the authentication of the copyrighted material; a step of extracting a watermark from said received copyrighted material; and a step of transmitting the authentication result to said user or copyright holder after comparing the copyright information obtained from said extracted watermark with stored copyrighted material and copyright information.

Hereinbelow, the service system and method for supplying/authenticating copyrighted material based on the watermarking technique according to the present invention will be more concretely explained with reference to the accompanying drawings.

Brief Description of the Drawings

Fig. 1 is a block diagram schematically illustrating the constitution of a system, which supplies a copyright authentication service on digital data according to the present invention.

Fig. 2 is a block diagram, showing an instance of the service supplying system according to the present invention, which illustrates the constitution of the system, being divided into server/client sides, for supplying a watermark embedding service.

Figs. 3A and 3B are flowcharts illustrating one embodiment of the operation

method in a service supplying system according to the present invention. Fig. 3A illustrates a process for embedding a watermark and Fig. 3B illustrates a process for extracting a watermark.

Fig. 4 illustrates an instance in case where a user is allowed to select a watermarking method during a process of Fig. 3A.

Fig. 5 is a flowchart schematically illustrating a process for embedding/extracting a watermark by using various watermarking methods provided by MDCAS.

Fig. 6 is a flowchart schematically illustrating a process in which a user is allowed to directly embed a watermark and to register the watermark-embedded content in MDCAS.

Detailed Description of the Preferred Embodiments

Fig. 1 is a block diagram, which schematically illustrates the constitution of a system for supplying a copyright authentication service on digital data according to the present invention. The system of Fig. 1 includes a user system (a service requester/a user; hereinafter referred to as 'client') 100 to be accessed via the internet, a multimedia data copyright authentication server (hereinafter referred to as "server") 120 for supplying Multimedia Data Copyright Authentication Service (hereinafter referred to as MDCAS); and a web browser 110 which plays a role as an intermediary of data communication between a DataBase Management System (hereinafter referred to as DBMS) 130 for storing information about the user information and various multimedia data, and a server 120 and a client 100. Said system may include a Payment Gateway 140 for surcharging on the use of the multimedia data.

According to the system of the present invention, the supply of copyrighted material of digital data or the copyright authentication is performed in the client 100 and the server 120 as a certificate authority of the digital data copyright. The above client 100 means all systems of copyright holders who allow users to utilize their copyrighted materials by making their copyrighted materials authenticated by means of the MDCAS or by registering their copyrighted materials in the database management system, as well as users who are supplied with the copyrighted materials by using the MDCAS.

Also, for the watermarking technique used in the above system, conventional various watermarking methods are available. Also, there are available methods taught by a watermarking-related patent application filed by the applicant of the present application (Korean Patent Application No. 98-37237: Watermarking method of digital image using wavelet conversion and discrete cosine conversion; Korean Patent Application No. 37274: Watermarking method of digital image using wavelet conversion and discrete cosine conversion).

In the present invention, "an extraction-based watermarking method" will be exemplified, which is used in the above-mentioned earlier patent applications and in which an imaged watermark is embedded into the original data and then extracted. In the case of image data, the extraction-based watermarking method capable of identifying the embedded watermark after extraction with the human eye has more commercial value, rather than "a detection-based watermarking method" in which a watermark is embedded by generating a conventional random sequence, i.e., a method of determining only the embedment of a watermark by calculating a co-relationship.

In other words, since the extraction method uses symbols, such as a trademark, logo, registered seal, emblem, or an autograph already publicly authorized or known to

the public, as an authentication means, the extraction method is much easier to protect and identify a watermark than the detection method using binary codes. However, the watermarking method is not limited to such method but various methods to be determined by a content holder are available as mentioned above.

In case that it is determined to be necessary to extract and authenticate copyright information for accurate evidence, the extraction-based watermarking method to be given as an example in the present invention may be used. However, for the case of determining only whether or not a watermark exists, the detection-based watermarking method may be used, in which a random bit sequence occurred by using a key (a specific key or user key) is used as a watermark. That is, there are various watermarking methods that have been developed or are presently being developed as described above, and thus a content holder can select a method suitable for his circumstance or business. However, such developed methods are quite different from each other and their purposes to be sought also considerably differ and thus, it is now difficult to mutually endorse such methods and manage a standardized content.

In order to overcome such problems, there is recently provided a two-layer digital watermarking method. Said two-layer digital watermarking method uses a mixture of a selective watermark (called as "real watermark") and a standard watermark (called as "meta watermark"), meaning a mixed use of a meta watermark and a real watermark.

For example, it means that specific indexes provided in each of the watermarking methods are embedded into a meta watermark of 8 bits (watermarking method for company A is No. 1, and that for company B is No. 2, etc.), information of each content or information of a copyright holder, etc., can be selectively and freely

embedded into a real watermark of 16 bits. That is, this method detects what watermarking technique is used from extraction of a meta watermark, and then, selects an extraction method therefrom to detect a real watermark. The present invention may also use the above two-layer digital watermarking method in selecting a watermarking method for an authorization service.

In the system according to the present invention, there are provided two services for supplying and authenticating a copyrighted material. As one service is embedding copyright information, it can simply embed copyright information into the original data as well as store in a certificate authority such information when a user registers the embedded data in the certificate authority and whether the copyright authentication (embedment of a watermark) is completed, thereby capable of issuing authentication data containing such information at any time when the user requests it. Also, this service makes it possible to find out a distribution channel of a copyrighted material by hierarchically embedding copyright information depending on whether a user is a copyright owner of the relevant copyrighted material or a simple user utilizing the copyrighted material.

The other service is authenticating copyright information of data by detecting and extracting a watermark from watermark-embedded data.

Examining the detailed operation of such services according to the foregoing basic characteristics, a client 100 connected to a server 120, the certificate authority, exchanges data through the web browser 110 on the internet. The server 120 receives the inputted specific data through the web browser 110 from the client 100, and then, stores the data in the database within the DBMS 130 or retrieves the database to transmit the requested data to the client 100 through the web browser 110. The

database within the DBMS 130 of the certificate authority is divided into a variety of tables, for example, user information, image profiles audio profiles and video profiles, etc. The user information table stores matters for proving a person's identity, such as a user's name, ID, password, address, identification number, as well as matters about payable means in the case of using a copyright authentication service (payment method, type of cards, card number, and term of validity, etc.).

Such user information table can be classified into copyright holder information containing a content copyright and content provider information. A person who wishes to be a member of the early stage should input the above-mentioned all his user information, which is stored in the database.

A registered member, whose all user information is stored in the database, should log in by using his ID and password. That is, in case where a registered member uses the authentication service in the future, the server 120 requests the DBMS 130 to verify the user information. Then, ID and password inputted by the user are transmitted to the database in the DBMS. In this case, according to the consistency or inconsistency of the information stored in the database and the information inputted, the result is transmitted to the server 120 and outputted to the user though the web browser 110.

Also, when a service is used, a service charge is paid by a registered payment means among the stored user information. For this purpose, in case where a user utilizes a service of the present invention, the authentication system of the present invention comprises a means for transmitting an itemized use statement to the payment gateaway 140 so that the user can pay his service charge. In addition, in order to provide the system of the present invention on the web, it is desirable that a united

platform interface such as an ActiveX control be provided to a user utilizing MDCAS.

For a watermark embedding method for providing copyrighted material for which a copyright is authenticated, there are provided a case where a watermark embedding process is performed by a user and an other case where the watermark embedding is performed by a certificate authority at a user's request. First, in the case of watermarking by the certificate authority, if a user transmits the original copyrighted material to the certificate authority (via on-line or off-line), then the certificate authority embeds a watermark image including a user's logo requested into the original copyrighted material and then transmits the watermark-embedded data to the user. Regardless of the place of watermarking, the user can distribute the watermarked data, not the original copyrighted material, on the internet.

For the performance of a watermarking by the certificate authority, a user should first register himself or his logo with the certificate authority. The processes of membership registration and logo registration will be needed in the future when providing a service for extracting a watermark from a watermark-embedded copyrighted material, and which are a registration procedure for authenticating a copyright. A logo of the present specification can be used as an identifier of a user such as an individual signature or a company's logo, and thus it is desirable that the logo be easily recognizable in the form of images. Therefore, it is desirable that a watermark to be embedded into a copyrighted material include a logo image together with additional information.

Likewise, in case where a user directly embeds a watermark, the user must register himself as a member along with registration of his logo. As stated previously, this is because such registration process bears a meaning as a procedure for registering a

copyright at the authenticate authority. A copyright authentication service for the case of a user's direct embedding of a watermark is also divided into to two cases. One is that both the generation of a logo and the embedment of a watermark are performed by a user. The other is that a logo is generated by the MDCAS and a watermark is embedded by the user.

In the first case, the MDCAS is not involved but a user performs both the generation of the logo and the embedment of the watermark. However, as stated above, even in the case that the user wishes the authentication of a copyright of the original copyrighted material by the certificate authority, the user must register the original, a logo, and data containing the copyright information in the certificate authority. In this case, the user can protect his copyright by the certificate authority by recording information about the entrance and registration dates in the certificate of authority.

In the second case, the client 100 generates his own logo by using the MDCAS server 120 and embeds it into the original copyrighted material, the constitution of which will be explained below with reference to Fig. 2.

Fig. 2 is a block diagram, as an example of a service supplying service according to the preset invention, which illustrates the constitution of a system divided into server/clients sides for providing a watermark embedding service. Unlike the constitution of Fig. 1, the constitution of Fig. 2 additionally includes a logo generation tool 220 enabling a user to generate a logo, and a MDCAS client 210 enabling a client 210 to operate the logo generation tool.

The following is the operation of the above constitution. First, the client 100 accesses the MDCAS server 120 by using a private ID and password given from becoming a member of the MDCAS server 120. A user requests the MDCAS server

120 to generate a logo to be used for his copyrighted material. The server 120 provides a tool for generating a new logo of the client 100, i.e., the logo generation tool 220. The logo generated therefrom is provided to the client 100 via the MDCAS client 210, simultaneously being stored in the database within the DBMS 130.

The client 100 receives the generated logo and then embeds it into the original copyrighted material by using the MDCAS client 210. The MDCAS client 210 existing between the client 100 and MDCAS server 120 is a kind of a virtual system model for a series of operation (watermark embedding operation) to be performed in the client 100. If the client 100 registers the logo in the certificate authority, then the MDCAS server 120 transmits to the client 100 messages as follows: "You should install a plug-in for using a watermark embedding service. Will you install?" If a user selects "YES" in response to this message, a program for the MDCAS client 210 is downloaded from the MDCAS server 120 and then installed in an information processing device of the client 100. If the capacity of the server 120 or its processing speed is excellent and thus there is no concern of extra-loading, it is possible to provide a watermark embedding service in the server without using the MDCAS client program.

Also, in case where a user simply requests the use of copyrighted material stored in the DBMS 130, he should make a request the use of a specific copyrighted material of the MDCAS server 120 without realizing the MDCAS client 210 or logo generation tool 220. Then, the MDCAS 120 embeds the previously registered user information as a watermark by using a watermarking technique of a predetermined method and then transmits the information embedded to the user, i.e., client 100. In this case, the information on the copyrighted material, such as information about a copyright holder and his logo, is embedded as a basic watermark and information about

a user who utilizes such copyrighted material, such as the time that the use of the copyrighted material is requested and personal information about the scope of the use and users, is embedded as a watermark. That is, information about a copyright holder and a user are hierarchically embedded, thereby it is possible to easily grasp a distribution channel of the copyrighted material.

According to another aspect of the present invention, a watermark is extracted from the watermark-embedded copyrighted material to verify and authenticate the copyright information, which is an essential step for a case where an illegal use of the copyrighted material is sensed and thus there arises a dispute between the original copyright holder and an illegal user. When such dispute arises, the certificate authority can authenticate a copyright of the original copyright holder by presenting evidencing materials such as "a certificate of authentication", which contains the registration date and information about a copyright holder and a copyright. At this time, if a user embeds a watermark, both the original data and doubtful data of illegal use are sent to the certificate authority for extracting a watermark embedded in the illegal data. The certificate authority authenticates a copyright after comparing the watermark extracted from the doubtful data with registered individual information and logo data.

In case that copyright information is embedded by the certificate authority, since there exist previously registered individual information and data such as original data and logo in the certificate authority, only the doubtful data of illegal use is sent to the certificate authority. Thereafter, the certificate authority extracts a watermark using the original stored in the database, and then, authenticates a copyright by means of comparison of the doubtful data with the copyright information such as a watermark and logo extracted.

Figs. 3A and 3B are flowcharts illustrating an embodiment of the operation method in a service supplying system according to the present invention. Fig. 3A illustrates a process of embedding a watermark and Fig. 3B illustrates a process of extracting a watermark. The basic flow in the above flowcharts is as follows: First, a user requests an access service of the certificate authority by using his registered private ID. If the ID is not registered, the user should take a step for registering the ID in the certificate authority. The user selects the type of service (copyright information embedment, copyright information extraction) and requests the corresponding service. The server transmits individual user information to DB, confirms it and executes the operation requested. Figs. 3A and 3B are flowcharts, which illustrate in detail the divided user/server sides of the MDCAS process supported on the internet web browser.

First, the process of embedding a watermark of Fig. 3A among the above entire processes will be reviewed. In order to use the copyright authentication service supplying system according to the present invention, a user accesses the internet to access a homepage providing the present service. Then, the user logs in a service system through a login browser which is outputted to the web browser (step 301).

User information is obtained from the database on the database management system 130 based on the user's ID and password inputted during the login process and then determines whether or not the user is registered (step 303). In the case of a new user, he inputs information on his personal information and payment method to be stored as the user information and then goes through the necessary steps to confirm the inputted information (step 305).

If the user is confirmed, the authentication service is begun. In the authentication service, a process of embedding/extracting a watermark and a data format

to be embedded as a watermark (for example, image size) are selected (step 307). If a service selected is a process of extracting a watermark, then it is proceeded to step 330. However, if it is a process of embedding a watermark, then it is checked whether the user is an existing service user (step 313). This is because in the case of an existing user, there exists a user logo previously registered and stored.

In the case of an existing service user, a user logo previously registered and stored in the database is transmitted to the user (step 325). In the case of a new service user, it is checked whether the user has a logo to be used as a watermark (step 315). If he has no logo to be used, a logo is generated by using the logo generation tool 220 provided in the system of the present invention and then a process for storing the generated logo in the client 100 and DBMS 130 is performed (step 317). When a logo is generated, target data in which a watermark is embedded is imported (step 321).

In the embodiment of Fig. 3A, a step of embedding a watermark into the target data is performed by a user (step 323). Concerning the watermark embedding, there may be presumed a case which enables a user to select a watermarking method. For example, in case where each of the watermarking companies (A,B,C) is operated together in the MDCAS server 120, the user may select one method among a variety of watermarking methods. Of course, the operation may be performed in the MDCAS server 120 by providing each watermarking algorithm to the MDCAS server 120 by each company. Otherwise, by operating the watermarking algorithm by each company, each company may perform the watermarking and the MDCAS may purely perform the authentication of the copyright.

In the above case, as illustrated in Fig. 4, the user logs in the MDCAS server 120 and goes through a basic process as mentioned previously. Thereafter, the user

selects one method among a variety of watermarking methods provided by the MDCAS server 120 (step 322), and then, embeds a watermark into data by using the selected watermarking method (step 323). Next, the embedded data and information thereof are stored and managed in the database within the database system 130 (step 326).

The watermark-embedded target data is transmitted back to a server of the certificate authority and the server stores the data received (step 327). Of course, in the step 327, it is judged whether the data to be stored is new data which needs to be newly stored. As a result, if the data is determined to be unnecessary, it is possible to omit them. Next, if a user outputs the watermark-embedded data and information on the watermark (step 329), the process of embedding a watermark is terminated (step 331).

In case that the operation judged from the step 309 is an extraction operation, the operation is performed according to the flow illustrated in Fig. 3B. During the extraction process, an objective data from which a watermark is to be extracted is imported (step 351). The user requests a watermark detection by transmitting the objective data to the server (step 353) and the server begins to perform a watermark detection process (step 355). The server detects whether there exists a watermark embedded during said process (step 357). If a watermark is detected, the watermark is extracted (step 361) and the server transmits user information about the extracted watermark and the extracted data to the user (step 363). However, if a watermark is not detected, a message to the effect that there does not exist a watermark to be extracted is outputted to the user (step 359), and the extraction process is terminated.

Briefly reviewing the foregoing process in which the user can select a watermarking method in connection with the extraction operation, as illustrated in Fig.

4, a user selects a company providing a watermark extraction method in a state that watermarked data are called (step 354). When the watermark extraction method is determined, a watermark is extracted from the corresponding data according to the determined method (step 355), and then, the authentication operation of the data is performed.

Also, in case where a watermark is embedded according to a watermarking method provided by MDCAS, not directly by a user, if it is necessary to authenticate multimedia data, information stored in the above MDCAS is called. Then, a watermark is extracted by using an extraction method of the same company that was selected when a watermark is first embedded. Thereafter, information about a series of copyright and content is authenticated from the extracted information.

Fig. 5 illustrates a method that enables a user to directly embed a watermark and register the watermark-embedded content in MDCAS. In this case, the user does not use the MDCAS but embeds a watermark in the content by a free method and with a product, which the user desires. However, if the user registers the watermark-embedded content in MDCAS for authenticating the content, the user transmits information about the watermarking method used together with the content. That is, such information contains that by what technique and products of which company the watermarking method is performed. The MDCAS stores said information together with the content. If the user requests a series of authentication, then the MDCAS extracts a watermark by selecting a watermark extraction method of each company from said stored information and authenticates it.

In addition, although the flowcharts of Figs. 3A and 3B do not display it, a user's itemized statement for payment is transmitted from the server to the payment

gateway 140 whenever the above watermark embedding/extracting services are performed. Its calculated result is transmitted to the server 120, and again directly to the user 100 or after adding up the total according to a given period.

In summary, the MDCAS server 120 stores, manages and provides data in which copyright information such as information about a copyright holder, the original data, and the copyright holder's logo is embedded. The MDCAS server 120 may also play a role as a provider providing the content.

As noted above, the copyright authentication service of multimedia data is generally explained. Said service is variously applicable according to the kinds of users utilizing such service and the forms of service. Here, the users utilizing the service are copyright holders, copyright authentication users said as sellers who sell copyrighted material, and purchasers who purchase copyrighted materials as authenticated. In this case, during the authenticating process, a user certificate authority (CA) for authenticating a user and a multimedia authentication certificate authority (MACA) for authenticating a copyright of a copyright holder or a seller are involved. All users have their own private keys and receive certificates of authentication enclosing public keys from the user certificate authority (CA).

First, for the case that a copyright holder uses a service according to the present invention, one embodiment of a service process provided to the copyright holder will now be explained. The copyright holder encrypts his original data with his private key. The copyright holder accesses the MACA with his personal ID and requests the copyright authentication of his data. At this time, the copyright holder transmits the encrypted data and a certificate of authentication to the MACA. The MACA first requests the CA to identify the user identity process about the copyright holder.

The CA confirms the data requested and then authenticates a user, i.e., a copyright holder. After authentication, the MACA converts the encrypted data to the original data by using the public key of the transmitted certificate of authentication. The MDCAS server in MACA retrieves a logo and user information previously registered by the user from the database and generates a watermark after organizing the information of the copyright holder retrieved from the database. The copyright information is embedded into the watermark. The logo and personal information (name, identity number, etc.) of the copyright holder transmitted to the server from the database and the logo of other certificate authorities are first embedded into the watermark, and automatically, the date and time information is inserted. Such process is called organization.

A watermark generated in such a method is embedded into the original data. The watermark-embedded data are transmitted to the copyright holder together with the certificate of authentication issued by the MACA. The original data, the generated watermark-embedded data and the copyright information are stored in the database within the DBMS. Further, in case where such process is performed, an itemized payment statement thereof is transmitted to the payment gateway.

In this case, it is possible for the MACA to sometimes play a role as a user certificate authority (CA) .

Next, in case where a user is a content seller, its service process is the same as that of the above process of a copyright holder. Since the seller has a duplication right of a relevant data copyright, he can be considered as being in the position of a copyright holder.

Finally, in case where a user is a content purchaser, the authentication process

between the CA, MACA and user will be explained. First, a purchaser purchases a digital content after accessing a site providing the content. In this case, the site providing the digital content authenticates the purchaser by using encryption algorithm and then provides the content to the authenticated user only. Thus, the content provider plays a role as the CA which authenticates a user. The MACA for a digital data authentication service according to the present invention performs the copyright authentication, operating with a digital content providing site, which performs a user authentication.

In more detail, the content purchaser first accesses the content providing site and is authenticated as a user after going through a registration process such as becoming a member of the site. The content provider, i.e. the CA, transmits to the MACA information about the content requested by the user (kinds of content, etc.), information about the content provider himself, and information about the purchaser. At this time, such information is transmitted in an encryption form.

The MDCAS server in the MACA receives such information and requests information and a logo about the content provider registered in the database of DBMS. The MDCAS server receives the information and logo requested from the database, makes a watermark to be embedded into the original data, and then, embeds the watermark into the original content data. The watermark-embedded data is encrypted again and then transmitted to the CA, i.e., the content provider, together with the certificate of authentication of the MACA. At the same time, the original data, the watermark-embedded data, and the copyright information are stored in the database of MACA.

The content provider transmits to the purchaser the watermark-embedded data

obtained from double-encryption of the received information by using a predetermined encryption method, which is agreed between a provider and purchaser. A right purchaser can obtain his desired content by using a double-encryption key previously provided by the content provider and encrypting the content.

In another embodiment of the present invention, watermark-embedded data may be transmitted directly to a purchaser, not to a content provider.

In such MACA-purchaser-CA (content provider) structure, in case that the MACA plays a role as a content seller, the function of the MACA and CA may be united into a single system.

In a process where the registration and authentication of a copyright using a watermarking system is performed, the present invention uses a new concept named "a copyright certificate authority" as above. The certificate authority is capable of copyright authentication as a third authority. Thus, it may be a single formal authority throughout the world and it may be established from a governmental point of view according to nations. Otherwise, an organization suitable for an establishment standard may be established according to a nongovernmental point of view so as to be authorized by a nation. In any case, since the internet is not confined to a national level but is a network spreaded throughout the whole world, a standard to be commonly used throughout the world is needed. Also, a certificate authority established suitable for such standard is needed and another higher organization is also needed to manage such certificate authority.

As above, in case that the authentication services of the same methods as those of Figs. 4~6 are performed, the different watermarking algorithm from each company may cause problems. As explained previously, since each of the watermarking

methods is quite different from each other in their purposes and methods, it is not easy for the MDCAS to manage them. Thus, in order to settle such problems, the two-layer digital watermarking method is used. This method provides each of the watermarking methods themselves with their own indexes, thereby making it possible to recognize which watermarking method is used and extract again information about a pure copyright and content based on such obtained information, and thus, it is very useful.

Industrial Applicability

As suggested above, the third certificate authority based on a watermarking technique should be established between a digital content provider and a user using the digital content for the following reasons: If both embedding and extracting methods of a watermark are disclosed to the general public, the confidence of copyright authentication would be decreased, and thus the third organization is needed for preventing the general public from performing either of embedding or extracting.

In case where an illegal use of a content is sensed, evidencing the embedment of copyright information by a watermarking system has the following effects: Content owners hope that an embedded watermarking would be a basis for argument of the preservation of his copyright as well as indisputable evidence of his copyright retention. In this case, the original data, copyright holder information, and data about an extracted watermark registered in a certificate authority having public confidence would be indisputable evidence to assert the copyright owner's rights. Thus, in the event that there arise multimedia data copyright-related disputes, such embedded-watermarking would be an influential basis capable of evidencing a copyright of the original copyright holder with respect to data.

Also, in case where each individual performs the embedment and extraction of a watermarking, it is not compelled to impose charges on the performance of embedment and extraction of the watermarking in a general sale of software. However, as suggested in the present invention, if it goes through the third authority, it is easy to impose fixed charges on the performance of use and extraction of a watermark.

The target data into which a watermark is embedded can be extended to data needed for copyright protection or information security such as important digital documents of a company, individual or governmental offices as well as such digital multimedia content. That is, by embedding secret documents or information into the existing multimedia data as a watermark, it is possible to prevent illegal disclosure of information to persons who are not allowed to read them.

The multimedia data copyright authentication authority according to the present invention performs the authentication of a copyright about certain digital copyrighted material as well as is applicable to the transmission/receipt of important documents in the electronic business on the internet. That is, a watermarking technique authenticated by the certificate authority is not confined to the function of simply embedding and extracting information but applicable to data hiding. Thus, if multimedia data such as generally used images or audio data are transmitted after hiding important document content, information, etc. in the same method as a watermarking technique, it is possible to reduce dangerous matters, such as files being illegally stolen, without using an existing encryption transmission method.

While there has been described and illustrated embodiments for multimedia data copyright authentication services according to the present invention, it will be understood to those skilled in the art that all modifications and conversion shall be

limited solely by the scope of the claims appended hereto.

What is claimed is:

1. A service system for supplying a copyrighted material, comprising:

a database for storing information about the copyrighted material; and

a copyrighted material supplying means for providing the copyrighted material, at a user's request through the internet to supply said copyrighted material, wherein said copyrighted material is received from said database and copyright information formed as a watermark is embedded into said copyright material.

2. A service system for supplying a copyrighted material, comprising:

a database for storing information about the copyrighted material; and

a copyrighted material supplying means for providing said copyright material, at a user's request through the internet to supply said copyrighted material, wherein the copyright information received from said database is embedded in a watermark form.

3. The service system for supplying the copyrighted material according to claim 1 or 2, wherein the transmission of data between said copyrighted material supplying means and said user is performed through a web browser.

4. The service system for supplying the copyrighted material according to any one of claims 1 to 3, wherein said watermark is embedded in an image form.

5. The service system for supplying the copyrighted material according to claim 4, wherein said copyright information includes information about a copyright holder and a copyright holder's logo.

6. The service system for supplying the copyrighted material according to claim 4, wherein a type of said watermark to be embedded is selected by said copyright holder.

7. The service system for supplying the copyrighted material according to claim 1, 2, 5 or 6, wherein said watermark is embedded utilizing a two-layer watermarking method.

8. The service system for supplying the copyrighted material according to claim 5 or 6, further comprising a logo generation means for generating a logo capable of indicating said copyright holder, wherein a logo generated by said logo generation means is stored in said database, via said copyrighted material supplying means, as said copyright information.

9. A service system for authenticating a copyrighted material, comprising:

a database for storing information about the copyrighted material; and

a copyrighted material authentication means for performing the authentication of the copyrighted material provided by a user, at said user's request through the internet to authenticate said copyrighted material, wherein a watermark is extracted from the copyrighted material received from said user; the extracted watermark is compared with information about said copyrighted material stored in said database; and the copyrighted material received from said user is authenticated.

10. The service system for authenticating the copyrighted material according to claim 9, wherein the transmission of data between said copyrighted material authentication means and said user is performed through a web browser.

Ex.1002 / Page 1017 of 1385
APPLE INC.

11. The service system for authenticating the copyrighted material according to claim 9 or 10, wherein said watermark is embedded in an image form.

12. The service system for authenticating the copyrighted material according to claim 11, wherein said copyright information includes information about a copyright holder and a copyright holder's logo.

13. The service system for authenticating the copyrighted material according to claim 11, wherein kinds of said watermark to be extracted are selected by said copyright holder.

14. The service system for authenticating copyrighted material according to claim 9, 10, 12, or 13, wherein said watermark is embedded utilizing a two-layer watermarking method.

15. A service method for supplying a copyrighted material, comprising the steps of:

registering a copyrighted material provided by a copyright holder through the internet and copyright information about said copyrighted material;

embedding said copyright information into said copyrighted material as a watermark at a user's request to provide a copyright with respect to said registered copyrighted material; and

providing said watermark-embedded copyrighted material to said user.

16. The service method for supplying the copyrighted material according to claim 15, wherein said watermark is embedded in an image form.

17. The service method for supplying the copyrighted material according to claim 16, wherein said copyright information includes information about a copyright holder and a copyright holder's logo.

18. The service method for supplying the copyrighted material according to claim 17, wherein a type of said watermark to be embedded is selected by said copyright holder.

19. The service method for supplying the copyrighted material according to any one of claims 15 to 18, wherein said watermark is embedded utilizing a two-layer watermarking method.

20. A service method for authenticating a copyrighted material, comprising the steps of:

receiving a copyrighted material at a user or copyright holder's request through the internet to authenticate the copyrighted material;

extracting a watermark from said received copyrighted material; and

transmitting the result of authentication to said user or copyright holder in comparison of the copyright information obtained from said extracted watermark with a stored copyrighted material and copyright information.

21. The service method for authenticating the copyrighted material, according to claim

20, wherein said watermark is embedded in an image form.

22. The service method for authenticating the copyrighted material, according to claim 21, wherein said copyright information includes information about a copyright holder and a copyright holder's logo.

23. The service system for authenticating the copyrighted material according to claim 21, wherein a type of said watermark to be extracted is selected by said copyright holder.

24. The service system for authenticating the copyrighted material according to claim 20, wherein said watermark is embedded utilizing a two-layer watermarking method.

25. A service system for transmitting and authenticating a copyrighted material, comprising:

a copyrighted material provider system which is operated by a provider of copyrighted material;

a database system for storing the copyrighted material provided by said provider of said copyrighted material;

a user system which is operated by a user; and

a server for authenticating the copyrighted material provided by and at a request of said copyrighted material provider and said user, and for providing the copyrighted material,

wherein said copyrighted material provider system is constructed in such a manner that the copyrighted material and copyright-related information for the

copyright authentication are provided to said server so as to be stored in said database system;

said user system is constructed in such a manner that the copyrighted material stored in said database system is selected and displayed at the request of the transmission of a specific copyrighted material; and

said server transmits, respectively, the information about said copyrighted material and the information about said user, which are embedded as watermarks, at a request from said user system to transmit the copyrighted material, and the watermark embedded in said copyrighted material is extracted so as to authenticate a copyright by comparison of the extracted watermark with the copyright-related information stored in said database system at said copyrighted material provider's request or said user's request to authenticate the copyright of the copyrighted material.

FIG 1. 1/6

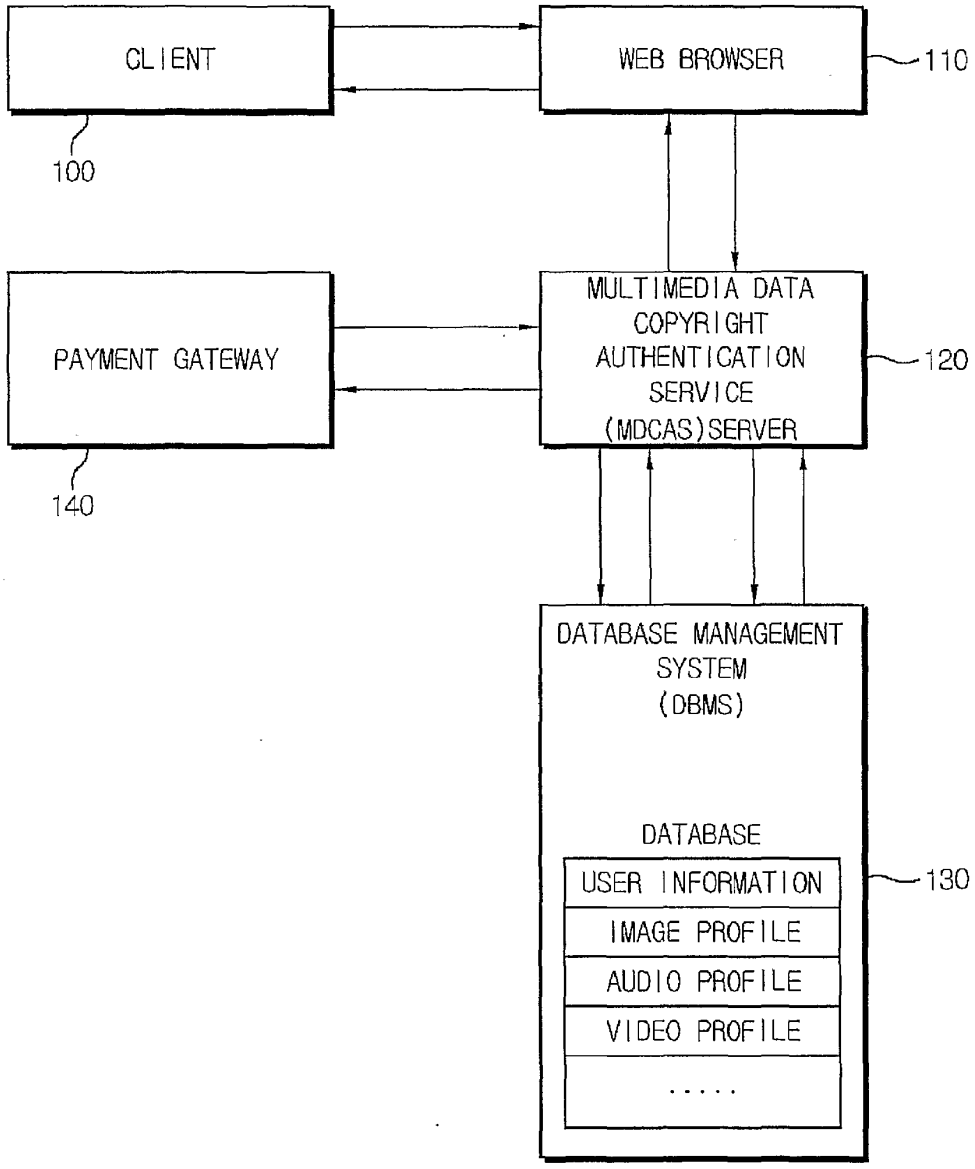

CLIENT
WEB BROWSER
110
100
PAYMENT GATEWAY
MULTIMEDIA DATA
COPYRIGHT
AUTHENTICATION
SERVICE
(MDCAS)SERVER
120
140
DATABASE MANAGEMENT
SYSTEM
(DBMS)
DATABASE
USER INFORMATION
IMAGE PROFILE
AUDIO PROFILE
VIDEO PROFILE
.....
130

FIG 2. 2/6

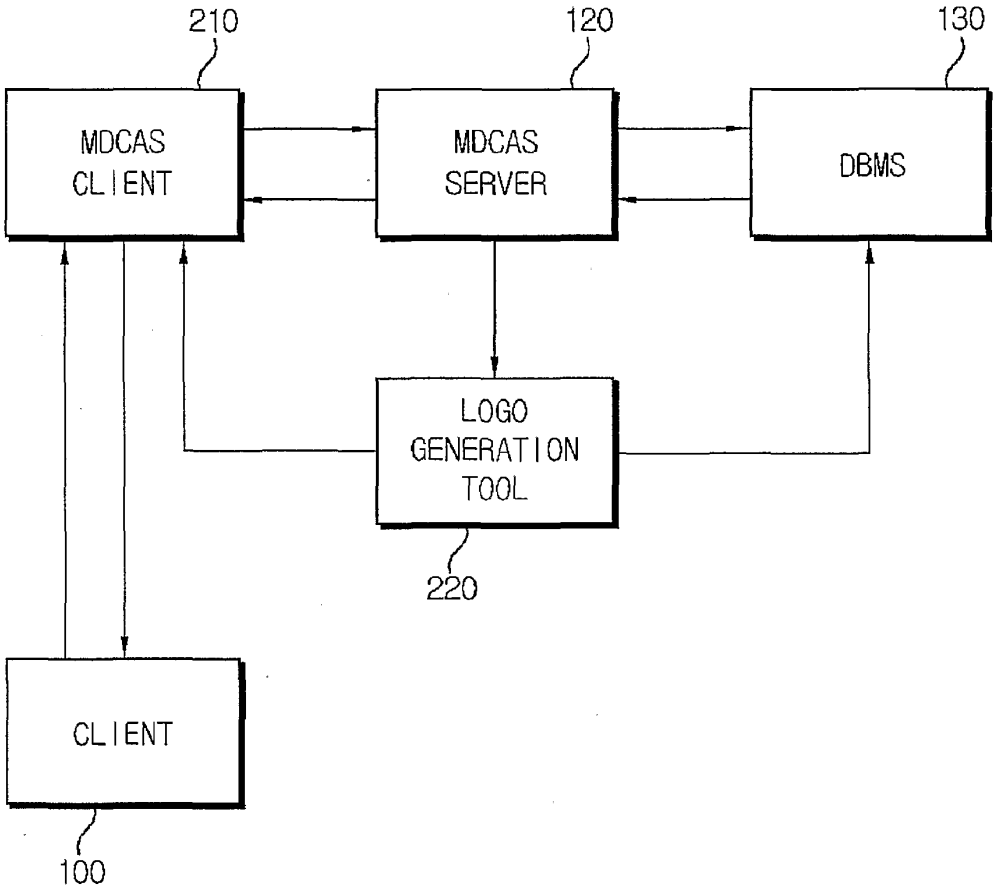

210
120
130
MDCAS CLIENT
MDCAS SERVER
DBMS
LOGO GENERATION TOOL
220
CLIENT
100

FIG 3a. 3/6

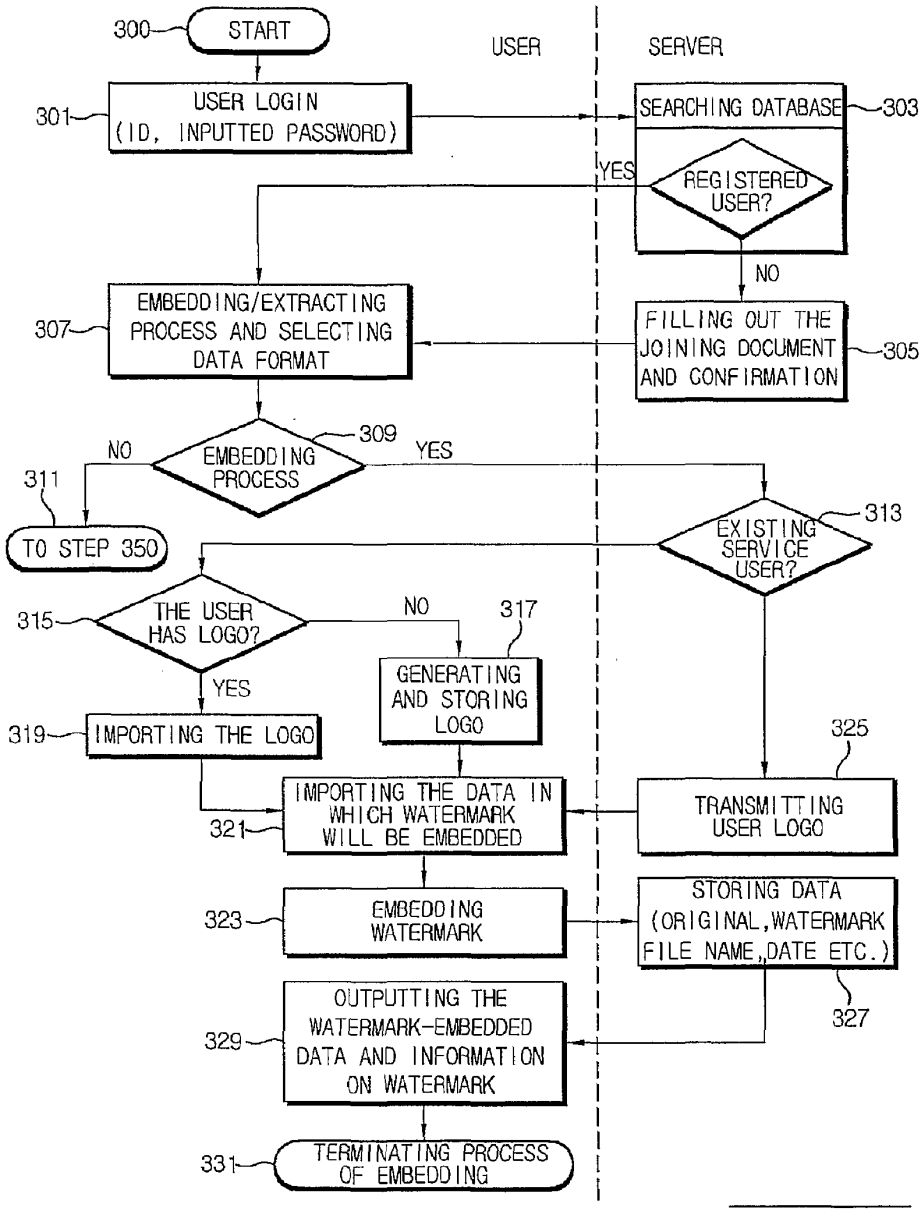

300
START
USER
SERVER
301
USER LOGIN
(ID, INPUTTED PASSWORD)
SEARCHING DATABASE
303
YES
REGISTERED USER?
NO
307
EMBEDDING/EXTRACTING PROCESS AND SELECTING DATA FORMAT
FILLING OUT THE JOINING DOCUMENT AND CONFIRMATION
305
309
NO
EMBEDDING PROCESS
YES
311
TO STEP 350
313
EXISTING SERVICE USER?
315
THE USER HAS LOGO?
NO
317
GENERATING AND STORING LOGO
YES
319
IMPORTING THE LOGO
325
IMPORTING THE DATA IN WHICH WATERMARK WILL BE EMBEDDED
321
TRANSMITTING USER LOGO
323
EMBEDDING WATERMARK
STORING DATA (ORIGINAL, WATERMARK FILE NAME, DATE ETC.)
327
329
OUTPUTTING THE WATERMARK-EMBEDDED DATA AND INFORMATION ON WATERMARK
331
TERMINATING PROCESS OF EMBEDDING

FIG 3b. 4/6

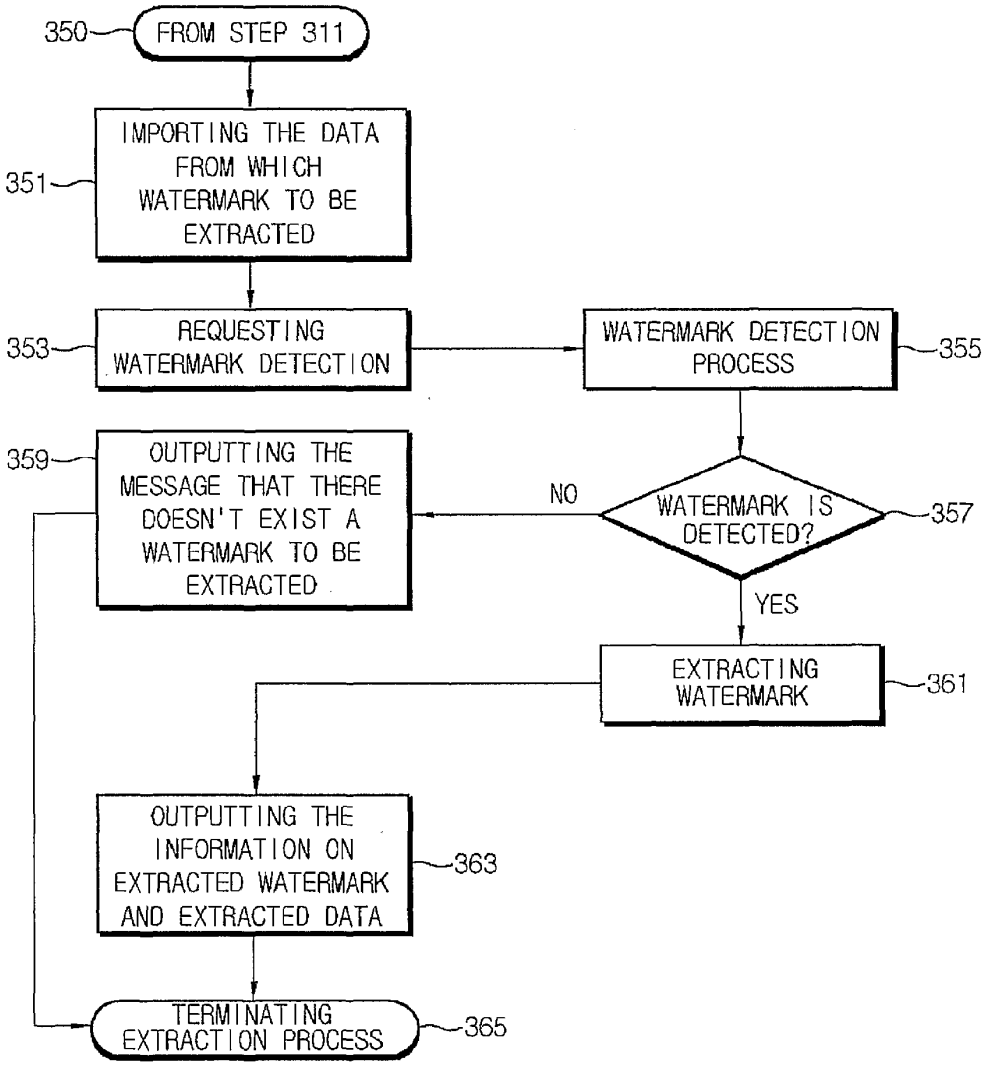

350
FROM STEP 311
IMPORTING THE DATA FROM WHICH WATERMARK TO BE EXTRACTED
351
REQUESTING WATERMARK DETECTION
353
WATERMARK DETECTION PROCESS
355
WATERMARK IS DETECTED?
357
NO
YES
OUTPUTTING THE MESSAGE THAT THERE DOESN'T EXIST A WATERMARK TO BE EXTRACTED
359
EXTRACTING WATERMARK
361
OUTPUTTING THE INFORMATION ON EXTRACTED WATERMARK AND EXTRACTED DATA
363
TERMINATING EXTRACTION PROCESS
365

FIG 4. 5/6

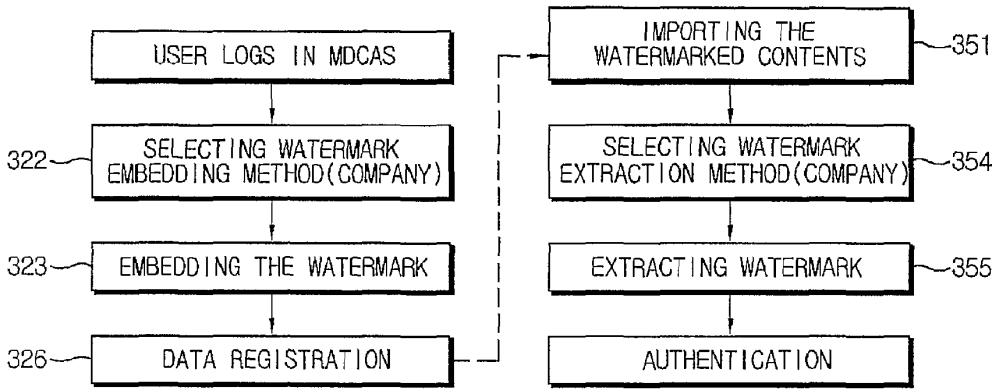
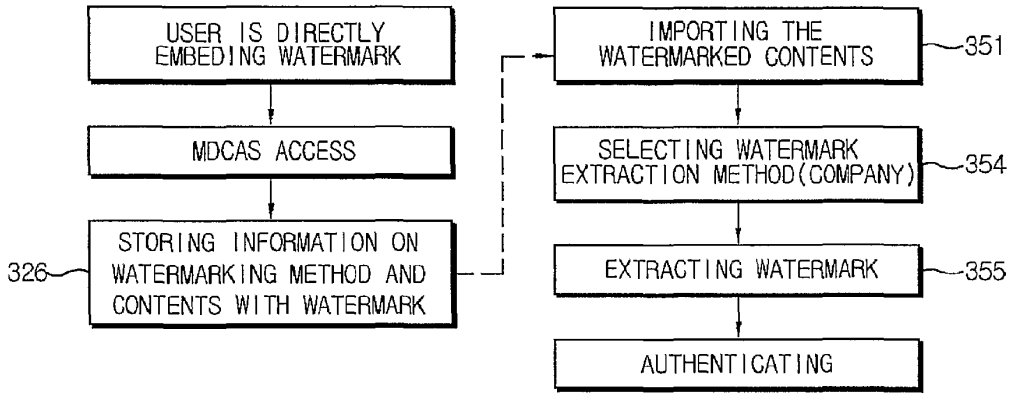

USER LOGS IN MDCAS
322
SELECTING WATERMARK EMBEDDING METHOD(COMPANY)
323
EMBEDDING THE WATERMARK
326
DATA REGISTRATION
IMPORTING THE WATERMARKED CONTENTS
351
SELECTING WATERMARK EXTRACTION METHOD(COMPANY)
354
EXTRACTING WATERMARK
355
AUTHENTICATION

FIG 5. 6/6


USER IS DIRECTLY EMBEDING WATERMARK
MDCAS ACCESS
STORING INFORMATION ON WATERMARKING METHOD AND CONTENTS WITH WATERMARK
326
IMPORTING THE WATERMARKED CONTENTS
351
SELECTING WATERMARK EXTRACTION METHOD(COMPANY)
354
EXTRACTING WATERMARK
355
AUTHENTICATING

INTERNATIONAL SEARCH REPORT

national application No.
PCT/KR01/00997

A. CLASSIFICATION OF SUBJECT MATTER

IPC7 G06F 17/60

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimun documentation searched (classification system followed by classification symbols)

G06T 9/00, H04N 5/913, G09C 5/00

Documentation searched other than minimun documentation to the extent that such documents are included in the fileds searched

KR, JP IPC as above

Electronic data base consulted during the intertnational search (name of data base and, where practicable, search trerms used)

WPI, JPO

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5875249 A (INTERNATIONAL BUSINESS MACHINES CO.) 23 FEBRUARY 1999	1-25
Y	KR 1999-82729 A (INTERNATIONAL BUSINESS MACHINES CO.) 25 NOVEMBER 1999	1-25
Y	KR 2000-18052 A (KIM JU HYUN) 6 APRIL 2000	1-25

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevence

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevence; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevence; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 JULY 2001 (30.07.2001)	31 JULY 2001 (31.07.2001)

Name and mailing address of the ISA/KR	Authorized officer
Korean Intellectual Property Office Government Complex-Daejeon, Dunsan-dong, Seo-gu, Daejeon Metropolitan City 302-701, Republic of Korea	JEON, Hyun Jin
Facsimile No. 82-42-472-7140	Telephone No. 82-42-481-5788

Form PCT/ISA/210 (second sheet) (July 1998)

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 February 2002 (07.02.2002)

PCT

(10) International Publication Number
WO 02/11446 A2

(51) International Patent Classification[7]: H04N 7/173

(21) International Application Number: PCT/EP01/07965

(22) International Filing Date: 11 July 2001 (11.07.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/627,188 27 July 2000 (27.07.2000) US

(71) Applicant: **KONINKLIJKE PHILIPS ELECTRONICS N.V.** [NL/NL]; Groenewoudseweg 1, NL-5621 BA Eindhoven (NL).

(72) Inventors: **MCGEE, Thomas**; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL). **DIMITROVA, Nevenka**; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL). **AGNIHOTRI, Lalitha**; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).

(74) Agent: **GROENENDAAL, Antonius, W., M.**; Internationaal Octrooibureau B.V., Prof Holstlaan 6, NL-5656 AA Eindhoven (NL).

(81) Designated States *(national)*: CN, JP, KR.

(84) Designated States *(regional)*: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 02/11446 A2

(54) Title: TRANSCRIPT TRIGGERS FOR VIDEO ENHANCEMENT

(57) Abstract: A system and method for retrieving information supplemental to video programming. Transcript text is searched for terms of interest and information associated with the terms is identified. Depending upon a user profile and the category of video segment being viewed, the supplemental information is formatted for display. Over time, the rules for associating the supplemental information with the terms of interest may be modified using a learning model.

Transcript triggers for video enhancement

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention is directed to the field of media technology. It is particularly directed to video and related transcript text.

2. Cross-Reference to Related Applications

This invention associates video with supplementary information using a text transcript, and extracts and augments textual features, as does co-pending application, Ser Nr. 09/351,086, Filed 1999 July 9 by the assignee, and incorporated by reference herein.

3. Description of the Related Art

In recent years, the number of media sources has increased and the volume of information from each source has also increased, resulting in information overload. Most consumers have neither the time nor the inclination to sift through the morass of information for what is pertinent to their wants and needs. Accordingly, so called "push technology" has developed. Webcasting applications such as Pointcast or Backweb, or the newer web browsers, ask the user which information categories and web sites the user is interested in. A web server then "pushes " information of interest to the user instead of waiting until the user requests it. This is done periodically and in an unobtrusive manner.

Concurrently, as media technology has progressed, the lines between video, audio, and other media have been blurred. Advances in media technology have enabled the delivery of Internet information and other informational material to the consumer's video display, along with the traditional television programming. Because the Internet has become a tool of e-commerce, consumers are conditioned to view a combination of media, video, audio, and text information on the same or associated topics. Consumers are acquainted with the hyperlink concept and the notion of "drilling down" to retrieve additional information on a subject they are viewing on the World Wide Web (WWW).

Retrieval of this additional information can currently be accomplished using closed caption text, audio, and automated story segmentation and identification. The Broadcast News Editor (BNE), provided by Mitre Corporation, enables such retrieval by automatically partitioning newscasts into individual story segments, and providing a

summary of each story segment in the first line of the closed-caption text associated with the segment. Keywords from the closed-caption text or audio are also determined for each story segment.

The Broadcast News Navigator (BNN), also from Mitre Corporation, sorts story segments by the number of keywords in each story segment that match search words selected by the consumer. Accordingly, story segments likely to be of interest to a particular consumer can be readily identified. However, using a combination of BNN and BNE requires that the consumer have an explicit search topic in mind, which is usually not the case in a typical channel-surfing scenario.

Patents which disclose providing the user with information supplemental to a television program include US Patent No. 5,809,471 to Brodsky entitled "Retrieval of additional information not found in interactive TV or telephony signal by application using dynamically extracted vocabulary" and US Patent No. 6,005,565 to Legall et al. entitled "Integrated search of electronic program guide, internet and other information resources." In the '471 patent, keywords are extracted from a television program or closed caption text, creating a dynamically changing dictionary. The user requests information based upon an item seen or word heard in the television broadcast. The user's request is matched against the dictionary, and when there is a match, a search for supplemental information to display is initiated.

In the '565 patent, the user selects topics and sources to search. Based on the user input, the search tool performs a search of the electronic program guide and other information resources such as the World Wide Web, and displays the results. Both the '471 patent and the '565 patent require that the user provide a keyword of interest. Neither patent relates the supplementary information retrieved to the global context of the program, (i.e. news program), as opposed to the subject matter of the program (i.e. the Stock Market report).

SUMMARY OF THE INVENTION

Accordingly, it would be advantageous to provide a method and system employing transcript text, for automatically providing supplementary multimedia information enhancing the consumer's television viewing experience. So called transcript text is comprised of at least one of the following: video text, text generated by speech recognition software, program transcripts, electronic program guide information, and closed caption text that contains all or part of the program information. Video text, is superimposed or overlaid

text displayed in the foreground, with the image as a background. Anchor names for example, often appear as video text. Video text may also take the form of embedded text, for example, a street sign that can be identified and extracted from the video image.

It would also be advantageous to provide supplementary information, which is specific not just to the individual consumer's known interests or profile, but also to the context of the program being viewed. For example, news segments would be associated with links to the Cable Network News (CNN) Web page while commercials would be associated with additional product information. The method and system would use learning models to continually develop new associations between the television content and other media content as well as to customize which type and how much supplementary information should be displayed. In this way, supplementary information would be integrated seamlessly with a television program without disturbing the viewer or requiring any action on the viewer's part.

The present invention addresses the foregoing needs by providing a system, (i.e., a method, an apparatus, and computer-executable process steps), for retrieval of supplementary information associated with a video segment, for display on the consumer's video display. The system includes a recognition engine for determining whether expanded keywords for retrieving supplementary information are contained in the closed captioned text accompanying the video segment or in other transcript related text. If a keyword is found, a stored rule indicates the supplementary information to be displayed, the information having been selected from a larger set of information, and selected in accordance with a user profile and the context of the segment. Alternatively, the transcript keywords are expanded and then matched to the user's profile. The context of the segment is automatically determined based upon classification data. These data include the program classification, object tracking methods, natural language processing of transcript information and/or electronic program guide information.

The information is displayed in a window or superimposed unobtrusively over the main video segment. Alternatively, the information is transmitted, for example to a hand-held device or an email account, stored to secondary storage, or cached in local memory. The system automatically recognizes the beginning and end of each segment, in the story classifications, and so is able to update the subset of rules to correspond to the program segment context.

In a further aspect of the invention, the set of rules for associating supplementary information with the video segment being viewed is dynamic and based upon a learning model. The set of rules is updated from a set of sources, including third-party

sources, and makes information available to the user in accordance with the user's choices and pattern of behavior. In one embodiment, the rules are transmitted from a Personal Digital Assistant (PDA) enabled with a wireless connection.

This brief summary has been provided so that the nature of the invention will be understood quickly. A more complete understanding of the invention is obtained by reference to the following detailed description of the preferred embodiments thereof in connection with the attached drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 depicts a system on which the present invention is implemented.

Figure 2 depicts elements of the processor contained within the system.

Figures 3a and 3b are flow diagrams used for explaining the operation of the present invention.

Figure 4 is a table illustrating supplementary information triggers for a given video segment, according to the present invention.

Figure 4a illustrates how keywords and triggers are expanded.

Figure 5 is a diagram of an embodiment of the invention illustrating a learning model.

Figure 6 is a diagram illustrating how the association rules database, for retrieving supplementary information, is updated and maintained.

Figure 7 is a diagram illustrating how supplementary information is displayed.

Figure 8 is a diagram illustrating one embodiment of the invention in which a set-top box is used.

Figure 9 is a diagram illustrating another embodiment of the invention in which a television display is used.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Figure 1 shows a representative embodiment of a system on which the present invention is implemented. In this embodiment, a multimedia processor system 6 includes a processor 12, a memory 10, input/output circuitry 8, and other circuitry and components well known to those skilled in the art. An analog video signal or a digital stream is input to the receiver 2. This stream is compliant with MPEG or other proprietary broadcast formats.

In accordance with the MPEG standard, video data is encoded using discrete cosine transform encoding and is arranged into variable length encoded data packets for

transmission. One version of the MPEG standard, MPEG-2 is described in the International Standards Organization — Moving Pictures Experts Group Document "Coding of Moving Pictures and Audio", ISO/IEC JTCI/SC29/WG11, July, 1996. MPEG is just one example of a format, which can be utilized in the system.

Transcript text, transmitted in the video signal 162, is extracted by the transcript extractor 4 from either line 21 of the analog video signal or the user data field of the MPEG stream. The transcript extractor 4 also partitions the video program into segments. The transcript text for the particular frame may be stored in the memory 10. Alternatively, it is analyzed as a real-time data stream.

Also stored in the memory 10 is Electronic Program Guide Information (EPG). This information, describing television broadcast information for a period of days or weeks, is downloaded on user request or at a preprogrammed time. It is transmitted by local analog TV broadcasters over the vertical blanking interval or through MPEG-2 private tables on a "home barker" channel. It can also be transmitted via telephone line or through wireless means. EPG data includes information such as the program's genre and subgenre, its rating, and a short program description. EPG data is used to determine the context of a program, such as whether it is a news program, a paid programming excerpt, a soap opera, or a travelogue.

Also stored in secondary storage 18 and available in the memory 10 is personal profile information, in the form of keywords or "triggers," describing the user's interests. Typical triggers could be "Clint Eastwood", "environment", "presidential election" or "hockey". These triggers are expanded in one aspect of the invention to include synonymous and related terms.

As is well known in the prior art, a personal profile of the user's interests is established automatically, by user input, or by a combination of both methods. For example, the TiVo™ Personal TV Service allows the user to indicate which programs the user prefers using a "Thumbs Up" or "Thumbs Down" button on the TiVo™ remote. TiVo™ then builds upon this information to select other related programs the user likes to view.

When a trigger matches keywords contained in the transcript text, supplementary data is retrieved, for example from the Internet 14 or proprietary sources 13 through the communication means 17. Another source for supplementary data is, for example, another channel. The data is then displayed to the user on a display 16 either as a Web page or a portion thereof or superimposed over the main video in a non-intrusive

fashion. Alternatively or additionally, a simple Uniform Resource Locator (URL) or informative message is returned to the viewer.

Rules for associating these triggers with supplementary data such as World Wide Web (WWW) pages are also stored in the secondary memory 18 and available from the memory 10. These rules are established through a default profile that is updated based on user behavior, or though a query program that prompts the user for interests and then generates the rule set. The rules are also received from a mobile device 15 such as a Personal Digital Assistant (PDA) or cell phone through the communications means 17. These rules associate supplementary information with the triggers, depending on the context of the program segment being viewed. For example, if a program segment is an advertisement for Clint Eastwood's new movie, the context is commercial and the supplementary data retrieved is a description of the movie he is starring in. If a program segment is a description of Clint Eastwood's car accident, the context is news, and the supplementary data retrieved is a biographical web page or a link to www.cnn.com to obtain more information about why he is in the news.

As illustrated above, association rules are also dependent upon a combination of EPG fields. For example, if "Clint Eastwood" appears in the actor's field of the EPG data, and the context is determined to be commercial, and the closed caption data is "We will be returning shortly to Clint Eastwood and Fist Full of Dollars after these announcements," then, the association rule retrieves supplementary data pertaining to the particular movie being shown. On the other hand, if "Clint Eastwood" does not appear in the actor's field of the EPG data, and the context is commercial, and the closed caption data is "High Plains Drifter starring Clint Eastwood will be aired on Friday," then, the association rule retrieves supplementary data pertaining to showtimes for the movie. These differences can be determined, for example, by comparing the text of the credits with text extracted from the closed caption data. It there is a match, then the program being advertised is the program being viewed. Alternatively, natural language processing can be used to identify key phrases such as "returning to" which would also indicate that the program being advertised is the program being viewed.

Alternatively, if "Clint Eastwood" does not appear in the actor's field of the EPG data, and the context is commercial, and the closed caption data says "Clint Eastwood's new movie will be released shortly", then the association rule retrieves supplementary data by linking to the Clint Eastwood home page to find out more about the movie.

Association rules also determine the category of media to be retrieved. For example, if "Kosovo" is the trigger and the program is sponsored by National Geographic, the association rule retrieves a map of the region. Alternatively, if the program segment context is news and the word "war" is located in the EPG data, then the association rule retrieves a recent political history of the region.

In alternative embodiments, the system includes a video display with built-in processing and memory, or a separate set top box for processing and storing information. These embodiments can include communication means or interface to communication means. Receipt of the video signal and Internet information is via wireless, satellite, cable or other media. This system is modifiable to transmit the supplementary information via the communication means 17 as an output signal over a radio transmitter, or via wireless means, where the signal is embodied in a carrier wave 160. The supplementary information is transmittable to an e-mail list, and/or downloadable to the voice mail feature of mobile devices 15 such as cell phones and/or transmittable to a hand held device such as the Palm Pilot®.

Figure 2 is a diagram of the processor elements. A profile generator 50 generates and stores a profile of the user's known interests, which includes trigger information or keywords of interest. This is accomplished for example through user input, by having the user respond to a series of queries, by creating a default profile based on user characteristics which are modified by the user, or by monitoring user activity to discover areas of interest. The rule generator 52 generates the association rules which logically combine each trigger with a variety of contexts to determine which supplementary information should be displayed to the user. The recognition engine 54 compares each trigger with the transcript text and determines whether the trigger exists as a keyword in the text. When a trigger is matched, the retrieving portion 56 retrieves the supplementary information and the formatting portion 58, formats the data for display. The context monitor 60, monitors the context to see whether it is changing due to the display of a new program segment. When a context change occurs, the context monitor 60 accesses the secondary storage 18 to retrieve a new subset of association rules.

The data updater 62 is used to update the supplementary information to incorporate new web sites, for example, or to reflect the results of searches performed by various search engines. The repetition counter 64 counts the frequency with which a particular piece of information is requested and the clickstream monitor 66 measures the frequency with which a user requests supplementary data in general. These intelligent agents

work in conjunction with the retrieval modifier 68 to modify the type of information and amount of information presented to the user.

Figures 3a and 3b are flow diagrams illustrating the method of the invention. To begin, in step S201, the input video is input to a receiver. The video is in analog or digital form. The transcript extractor, which is separate from or incorporated into the processor, extracts the transcript text in step S202 and identifies the beginning and end of each video segment. Next, in step S203, the processor retrieves the keywords from the transcript text. Extraction of keywords is well known in the art and one such method of extraction is described in U.S. Patent No.5,809,471 to Brodsky, entitled "Retrieval of additional information not found in interactive TV or telephony signal by application using dynamically extracted vocabulary." As shown in Figure 4a, these keywords 152 are extracted from the transcript text 150 and expanded 154 to achieve more meaningful and complete results, by associating them with synonymous or related keywords as shown in Figure 3a step S204. A thesaurus is used for this purpose or a database such as Wordnet®. Wordnet® is an on-line lexical reference system whose design is inspired by current psycholinguistic theories. The various parts of speech are organized into synonym sets, each representing one underlying lexical concept.

Keywords can also be expanded by identifying the theme of the transcript text. For example, the presence of the trigger “economy” in transcript text can be derived, when a number of words such as “inflation”, "Alan Greenspan", and "unemployment rate" are simultaneously present. Similarly, the presence of the trigger "President Clinton" can be derived if the keyword "President of the United States" is present in the transcript text.

Special rules apply when the supplementary data is contained in reference tools such as dictionaries and encyclopedias, as shown in Figure 4 114 132. In one mode, triggers are mapped to a variety of keywords depending on the level of understanding of the viewer. For example, if the viewer is a child or foreign-speaking viewer, the trigger "unemployment" would be mapped to the keyword phrase "without a job" but would not be mapped to the keyword "redundancy." In an alternate mode, the keywords are expanded as described above.

Parental control is implemented below the program level at the program segment or contextual level. Therefore, parents need not worry if a commercial inappropriate for children is shown during an otherwise appropriate cartoon show, for example. The child viewer is presented with a special screen only during the commercial. This special screen may take the form of a toy advertisement instead of merely a typical blocking screen.

Blocking triggers are also expanded to enhance the effectiveness of the blocking. For example, if the parent does not want the child to see video segments related to war, the trigger "war" is mapped to keywords and phrases such as "armed conflict" and "bombing." An example of trigger expansion is shown in Figure 4a 102 156.

Returning to Figure 3a, in step S205, the personal profile containing the triggers is read. The processor matches the keywords developed from the transcript text with the triggers contained in the user profile in step S206. If there is no match, the processor continues by extracting additional transcript text.

If there is a match, in step S207 of Fig. 3b, the context of the ongoing video program is identified. This is done in several ways, using either the closed caption data, EPG data, object tracking methods, or low-level feature extraction such as color, motion, texture, or shape. The context of the program segment is also extracted from the transcript text using natural language techniques. For example, Microsoft Corporation has developed software that learns by analyzing existing texts, including online dictionaries and encyclopedias, and automatically acquiring knowledge from this analysis. This knowledge is then used to help constrain the interpretation of the word "plane" in a sentence like "Flying planes can be dangerous" and to determine that the sentence pertains to aviation rather than woodworking.

Software also operates at the discourse level, using discourse analysis to identify the structure of the closed caption text and thereby its context. For example, a news program is identified because it would generally report the most important facts, "who, what, when, where, how" in its beginning. Accordingly, a program that began with the sentence "Clint Eastwood was in a gun fight, in Carmel California, at seven a.m. on Main Street, by a bystander with a home video camera" is identified as a news story. The context is also available in the EPG data from the genre and sub genre fields or a combination of fields as explained above.

Next, in step S208, the association rules are read. The association rules determine which supplementary data from a stored database should be retrieved, based upon the keyword and context. In step S209, the customized display modules are read. These modules enable the user to restrict the types of information, and therefore also the amount of information, the user wants to view. For example, the user may only wish to see the Uniform Resource Locator (URL) of a WWW page, only larger titles from the page, a page summary, or a full page. The user can choose the supplementary sources he wants to view and prioritize these sources.

In step S210, the supplementary data is retrieved from a database stored in memory. The database contains items of interest or pointers to items of interest, ancillary to the trigger. For example, the database contains any of the following: names of celebrities and public figures, geographic information such as countries, capitals, and presidents, product and brand names, assorted categories and topics.

The database is maintained and refreshed from an established set of sources. These include for example, the Bloomberg site, encyclopedias, thesauri, dictionaries, and a set of web sites or search engines. Information from the EPG and closed caption data is also incorporated into the database.

A set of refresh and cleanup rules, as shown in Figures 5 and 6 is also stored in a database or a viewer's profile, for example, and maintained for managing the size of the database or profile and its currency. For example, "stale" items such as election results and links to information about polls and the candidates would be deleted after an election takes place.

Returning to Figure 3b, in step S211, the supplementary information is formatted for display. The information is displayed in a window or superimposed unobtrusively over the main video segment. Alternatively, the information is formatted for transmittal, for example to a hand-held device such as the Palm Pilot™ distributed by Palm, Inc. or to an email account.

Figure 4 illustrates the set of association rules 100 for several triggers 102. In the table, the first column represents the triggers 102 and columns 2-4 represent the possible contexts 104, 106, 108, 110 for the example triggers shown. Beginning with the association rule 120 for the first trigger 102, "Clint Eastwood", when this trigger 102 appears in a user's profile, one of three different items of supplementary information 116, 118, 120 are retrieved for display, depending on the context in which Clint Eastwood appears in the video segment being viewed. Although only one link is shown in each box of the example table, multiple links can exist. If Clint Eastwood appears in a commercial, the system will link to the WWW page located at www.imdb.com and display the page in accordance with the customized display model. If Clint Eastwood appears on a talk show, the talk show segment where he appears will be stored for retrieval 118 and/or an alert sent to the viewer in real-time. Alternatively, an offline alert is transmitted for later viewing, notifying the viewer that the segment has been stored.

Alerts are automatically or manually retrieved. Alert transmission is also keyed to a topic such that the alert is displayed the next time a Clint Eastwood movie is

shown. If Clint Eastwood appears on a news program, the system will link to the WWW page located at www.cnn.com. Alerts have priorities enabling the user to select the circumstances when the user wants to be notified. For example, a user may only want to view alerts pertaining to severe weather warnings.

The second association rule 122 for the trigger 102 Macedonia deals with 4 different contexts. If the trigger "Macedonia" appears in an advertisement, the system links to the WWW page at www.travel.com 130. If Macedonia is the subject of a talk show, the system links to an entry for "Macedonia" in Compton's Encyclopedia 132. If Macedonia is the subject of a news show, the user is tuned to the station where the program is being aired 134. If Macedonia is the subject of a program sponsored by National Geographic magazine, the system links to www.yahoo.com/maps 136 to display a map of Macedonia.

Association rules 3-5 124 126 128 should be interpreted in the same manner as the above examples. As shown in the table, when certain triggers 102 such as "Meryl Streep" appear in transcript text, the system will only provide supplementary information for certain contexts. In the case of "Meryl Streep", supplementary information is only supplied for the Talk Show and News contexts. If desired, such a rule is broadened to apply to a list of well-known actors or all actors.

Figure 4a illustrates how both the triggers and keywords can be expanded to retrieve supplementary information. For the example transcript text 150 shown, the keyword 152 "Lyme Disease" is extracted from the transcript text 150. The keyword 152 is then expanded to map to the additional key words "tick", "tick bite", "bull's eye rash" and "deer tick." If any of these expanded keywords appear in the transcript text, supplementary information related to Lyme Disease will be retrieved.

Figure 4a also illustrates how triggers are expanded. The trigger 102 "Lyme Disease" is expanded 156 to include the related terms "tick bite", "West Nile virus, and "mosquito spraying." Accordingly, if the transcript text 150 contains any of the expanded triggers the segment is stored, for example.

Figure 5 illustrates how a learning model is implemented to continually update the customized display modules and association rules. The repetition counter 20 maintains a count of how often the user requests the same supplementary data, for example by clicking on a URL. Also, more than one piece of supplementary information may be retrieved by the retrieving portion 56 of the processor, shown in Figure 2, for each segment and the user may select the information the user wishes to view. If a user requests a particular piece of supplementary data less than a predetermined amount of times, the stored association rules 26

are updated by the retrieval modifier 24 such that the supplementary data is eliminated from the rule or the rule is modified to include a new source. The clickstream monitor 22 monitors how frequently the user requests any supplementary data. If the user selects supplementary data less than a predetermined amount of times, the custom display module 28 for that user is modified by the retrieval modifier 24 such that less information is presented to the user.

Figure 6 illustrates how the dynamic association rules database is updated and maintained. The database contains items of interest or pointers to items of interest that can provide ancillary information, when triggered by a match between a keyword in the transcript text and a trigger in the user's profile. The database is updated over time to reflect current events and to match the evolving user profile.

The existing data sources set 36, specifies the data sources from which the association rules database 26 is constructed. The data sources set 36 which includes both external data 38 from a variety of published sources, proprietary information, and data from the Internet 14 is updated by the data updater 40 to incorporate new web sites, for example, or to reflect the results of searches performed by various search engines. A set of refresh rules 32 is maintained to keep the size of the database at a preset limit. According to a set of established priorities, information is deleted when necessary. A set of cleanup rules 34 is also maintained which specify when and how "stale" information can be deleted. Information in certain categories is date stamped, and information older than a preset number of months and/or years is deleted.

Figure 7 illustrates an embodiment in which the supplementary information 70 is displayed superimposed unobtrusively over the main video segment. The supplementary information appears at the bottom of the picture.

Figure 8 illustrates an embodiment in which a set-top box 75 comprises a receiver 2, which receives video program and transcript text. A transcript text extractor and segmenter 4 extracts the transcript text 150 from the video signal and associates it with segments of the video program such as commercials and news flashes. A processor system 6 includes processing elements well known in the art -- an input/output portion 8, a memory 10, and a processor 12. Via a communication means 17, the processor system retrieves information supplemental to the video program from a variety of sources. Three of these sources, the Internet 14, proprietary (non-public) databases 13, and mobile devices 15 such as PDAs are shown in the figure as examples. The communication means 17 can connect to other devices not specifically shown, via wireless means, cable modem, a digital subscriber line, or a network, for example. The secondary storage 18 is used to store the supplementary

information as well as the rules for retrieving the information. The set-top box can be interfaced to a display such as a PC display or a television.

Figure 9 illustrates another embodiment in which a television 80 comprises a receiver 2, a transcript text extractor and segmenter 4, a processor system 6, secondary storage 18, a communication means 17, and a display 16. The processor system 6 includes processing elements well known in the art - - an input/output portion 8, a memory 10, and a processor 12. The television 80 interfaces to sources of supplementary information via the communication means 17 which interfaces to the Internet 14, proprietary sources 13 and mobile devices 15, for example.

The present invention has been described with respect to particular illustrative embodiments. It is to be understood that the invention is not limited to the above-described embodiments and modifications thereto, and that various changes and modifications may be made by those of ordinary skill in the art without departing from the spirit and scope of the appended claims.

CLAIMS:

1. An association method for retrieving information supplemental to a video program comprising the steps of:

receiving the video program (2);

identifying in the video program at least one segment (4);

receiving classification data for said at least one segment (4,2);

receiving transcript text for the video program (4);

identifying a user profile for a video program viewer (50);

identifying a set of rules (52) incorporating the classification data, for associating the supplementary information with the video program, when the transcript text and the user profile satisfy a set of conditions; and

automatically retrieving the supplementary information based upon the set of rules for display on a display (56).

2. The method according to Claim 1, wherein the set of rules (100) includes information from the user profile (102).

3. The method according to Claim 2, wherein the user profile contains at least one trigger (102) which identifies a topic of interest to the video program viewer.

4. A method according to Claim 3, wherein the set of conditions specifies that a recognition engine (54) retrieve the supplementary information only when a keyword in the transcript text matches (S206) the at least one trigger (102) in the user profile.

5. The method according to Claim 1, wherein the transcript text is comprised of closed caption text, video text, program transcripts or electronic program guide information.

6. The method according to Claim 1, wherein the transcript text (150) is generated by speech recognition software.

7. The method according to Claim 1, further including the step of receiving at least a portion of the set of rules (100) from a mobile device (15) or a third-party source (13).

8. The method according to Claim 1, wherein at least part of the supplementary information and pointers to the supplementary information are stored in a database (26) or transmitted to a personal digital assistant (15) or to an electronic mail address (14).

9. The method according to Claim 1 wherein the retrieval of the supplementary information (116,118,120) is in real-time.

10. The method according to Claim 1, wherein the supplementary information (116,118,120) is formatted for display in a window (70) or for superimposition over the video program on a display (16).

11. The method according to Claim 1, wherein the supplementary information is text information (114) or a page from the World Wide Web (116).

12. The method according to Claim 5, further including the step of automatically selecting the set of rules (100) for each video program segment from the electronic program guide information (150).

13. The method according to Claim 3, further including the step of automatically selecting the set of rules (100) by applying natural language processing to the transcript text (150) for each video program segment to identify whether a keyword (S203) in the transcript text (4) matches a trigger (102) in the user profile.

14. The method according to Claim 3, further including the step of identifying at least one keyword (S203, 152) in the transcript text (150), expanding the at least one keyword (S204, 152) to include related terms (154), and retrieving the supplementary information (S210) when the keyword or related terms matches (S206) the at least one trigger (102) in the user profile.

15. The method according to Claim 3, further including the step of automatically generating the set of rules (52) by applying discourse analysis to the transcript text (150) for

each video program segment to identify whether a keyword (152) in the transcript text (150) matches a trigger (S206,102) in the user profile.

16. The method according to Claim 3, further including the step of expanding at least one trigger (154) in the user profile to include related terms, identifying at least one keyword in the transcript text, and retrieving the supplementary information when the trigger or related terms matches the at least one keyword in the transcript text.

17. The method according to Claim 8, further including the step of deleting (40) supplementary information (26) or pointers to supplementary information added to the database before a certain date or related to events that have terminated.

18. The method according to Claim 11, wherein only the Uniform Resource Locator (URL) (28,70) of the page or wherein a portion of the page (28) which is less than the entire page or wherein a summary of the page (28) is displayed.

19. The method according to Claim 1, further including the step of monitoring (22) the amount of supplementary information viewed by the video program viewer, and the frequency (20) with which the video program viewer views the supplementary information, and varying (24) the amount of supplementary information formatted for display correspondingly, according to a predetermined formula.

20. The method according to Claim 1, wherein the supplemental information is included in an electronic mail message (15) or is downloaded (17) to a personal information manager (15).

21. An apparatus for retrieving information supplementary to a video program, the apparatus comprising:

a receiver (2) which receives the video program, classification data for the video program, and transcript text for the video program;

a transcript extractor (4) which identifies at least one segment within the video program and associates transcript text with said one segment;

a context monitor (60,S207), which monitors the classification data (104,106,108,110) for each segment thereby identifying a context for each segment;

a profile generator (50), which establishes a user profile for a video program viewer;

a rule generator (52), incorporating the classification data (102,104,106,108,110), which establishes a set of rules (100) for associating supplementary information (116,118,120) with the video program, when the transcript text (150) and the user profile (102) satisfy a set of conditions;

a retrieving portion (56), which retrieves the supplementary information (116,118,120), based upon the set of rules (100);

a formatting portion (58) which formats (S211) the retrieved supplementary information for display along with the video program.

22. An apparatus according to Claim 21 wherein the retrieving portion retrieves (S210) the supplementary information (116,118,120) when a trigger (102) within the user profile matches (S206) a keyword (152) within the transcript text.

23. An apparatus according to Claim 22, wherein at least one trigger (102) in the user profile is expanded (156) to include related terms and the trigger and the related terms are compared (S206) with the keyword (152).

24. An apparatus according to Claim 22, wherein at least one keyword (152) within the transcript text (150) is expanded (154, S204) to include related terms and the trigger (102) is compared with the keyword (154) and the related terms.

25. An apparatus according to Claim 21, wherein the retrieving (S207, 104,106,108,110) portion (56) retrieves information for the segment based upon the context of the segment.

26. Computer-executable process steps to retrieve information supplemental to a video program, the computer-executable process steps being stored on a computer-readable medium (18) and comprising:

a receiving step (S201) to receive the video program, classification data describing the video program, and transcript text for the video program;

a context identifying step (S207) to identify at least one segment in the video program and the context of the segment based upon the classification data;

a keyword identification step (S203) to identify keywords in the transcript text for the at least one segment in the video program;

a keyword expanding step (S204) to expand the keywords to include related terms;

a personal profile retrieving step (S205) to retrieve a user profile for a viewer viewing the video program;

a keyword matching step (S206) to match the keywords and the related terms with the at least one trigger in the user profile;

an association rules retrieving step (S208) to retrieve a set of rules specifying which information supplemental to the video program will be retrieved, depending upon the identified context;

a retrieving step (S210) to retrieve the supplementary information based upon the set of rules when the keyword matching step is successful; and

a formatting step (S211) to format the retrieved supplementary information for display;

27. A signal (160), embodied in a carrier wave, representing a video program (162) and information supplemental thereto (116,118,120), comprising video program classification data (104,106,108,110); transcript text (150); a user profile (102); and rules (100) incorporating the video program classification data, for associating the supplementary information with the video program when the transcript text and the user profile satisfy a set of conditions (S206).

28. An apparatus for retrieving and displaying information supplemental to a video program comprising:

means (2) for receiving the video program (162);

means for identifying in the video program at least one segment (4);

means for receiving program classification data describing the at least one segment (4,2);

means for receiving transcript text (150) for the video program and associating the transcript text with the at least one segment (4);

means for retrieving a user profile for a video program viewer (50);

means for identifying (52) a set of rules (100), incorporating the classification data (104,106,108,110), for associating the supplementary information (116,118,120) with

the video program, when the transcript text and the user profile (102) satisfy a set of conditions (S206);

means for retrieving the supplementary information based upon the set of rules (56,S210); and

means for formatting (58) the supplementary information for display along with the video program.

29. A set-top box (75) for a video program viewer, comprising:

receiving means (2) which receives a video program (102), classification data for the video program (104,106,108,110), and transcript text (150) for the video program;

transcript text extraction and segmenting means (4) which identifies at least one segment in the video program and associates transcript text with the at least one segment;

communication means (17) which connects to at least one information source (14,13,15) and receives information supplemental to the video program (116,118,120);

processor means (6) which

a) retrieves a user profile (50) for the video program viewer which contains at least one trigger (102) reflecting an interest of the video program viewer,

b) associates the classification data with the at least one segment (60, S207),

c) identifiifes a set of rules (52) incorporating the classification data, for associating the supplemental information with the segment,

d) searches the transcript text for a trigger contained in the user profile (54),

e) retrieves the supplemental information (56), using the communication means (17) and based upon the set of rules (100), when the trigger (102) is contained within the transcript text (150), and

f) formats (58) the retrieved supplemental information for display; and

storage means (18) which stores the transcript text, the user profile, the set of rules, and the supplemental information.

30. The set-top box (75) according to Claim 29, wherein the receiving means receives a digital video program.

31. The set-top box according to Claim 29 (75), wherein the processor (12) decodes and formats the digital video program for display on an analog display.

32. The set-top box (75) according to Claim 29, wherein the video program viewer selects a destination (15) where the supplementary information will be transmitted via the communication means (17).

33. The set-top box (75) according to Claim 29, wherein more than one type of supplementary information (116,118,120) is retrieved by the processor (12) for each segment, the retrieved supplementary information is automatically placed in an order of priority according to the user profile (S209), and the supplementary information with highest priority is formatted for display (S211) by default.

34. The set-top box (75) according to Claim 29, wherein more than one type of supplementary information (116,118,120) is retrieved by the processor (12) for each segment, and the video program viewer selects the retrieved supplementary information the video program viewer wishes to view.

35. A television set (80) comprising:

receiving means (2) which receives a video program (162), classification data for the video program (104,106,108,110), and transcript text (150) for the video program;

transcript text extraction and segmenting means (4) which identifies at least one segment in the video program and associates transcript text with the at least one segment;

communication means (17) which connects to at least one information source and receives information supplemental to the video program;

processor means (12) which

a) retrieves a user profile (50) for a video program viewer which contains at least one trigger reflecting an interest of the video program viewer,

b) associates the classification data with the at least one segment (4,2),

c) identifies a set (52) of rules (100), incorporating the classification data, for associating the supplemental information with the segment,

d) searches the transcript text (54) for a trigger (102) contained in the user profile,

e) retrieves the supplemental information (116,118,120), using the communication means (17), and based upon the set of rules (100), when the trigger (102) is contained within the transcript text, and

f) formats (58) the retrieved supplemental information for display;

storage means (18) which stores the transcript text, the user profile, the set of rules, and the supplemental information; and

display means which displays the video program and the retrieved and formatted supplemental information.

36. Computer-executable process steps to retrieve information supplemental to a video program, the computer-executable process steps being stored on a computer-readable medium (18) and comprising:

a receiving step (S201) for receiving the video program, classification data describing the video program and transcript data for the video program;

a segmenting step (S202) for identifying at least one segment in the video program and classification data for the segment;

a first identifying step (S205) for identifying a user profile for a video program viewer;

a second identifying step (S208) for identifying a set of rules incorporating the classification data, for associating the supplementary information with the video program, when the transcript text and the user profile satisfy a set of conditions; and

a retrieving step (S210) for automatically retrieving the supplementary information based upon the set of rules.

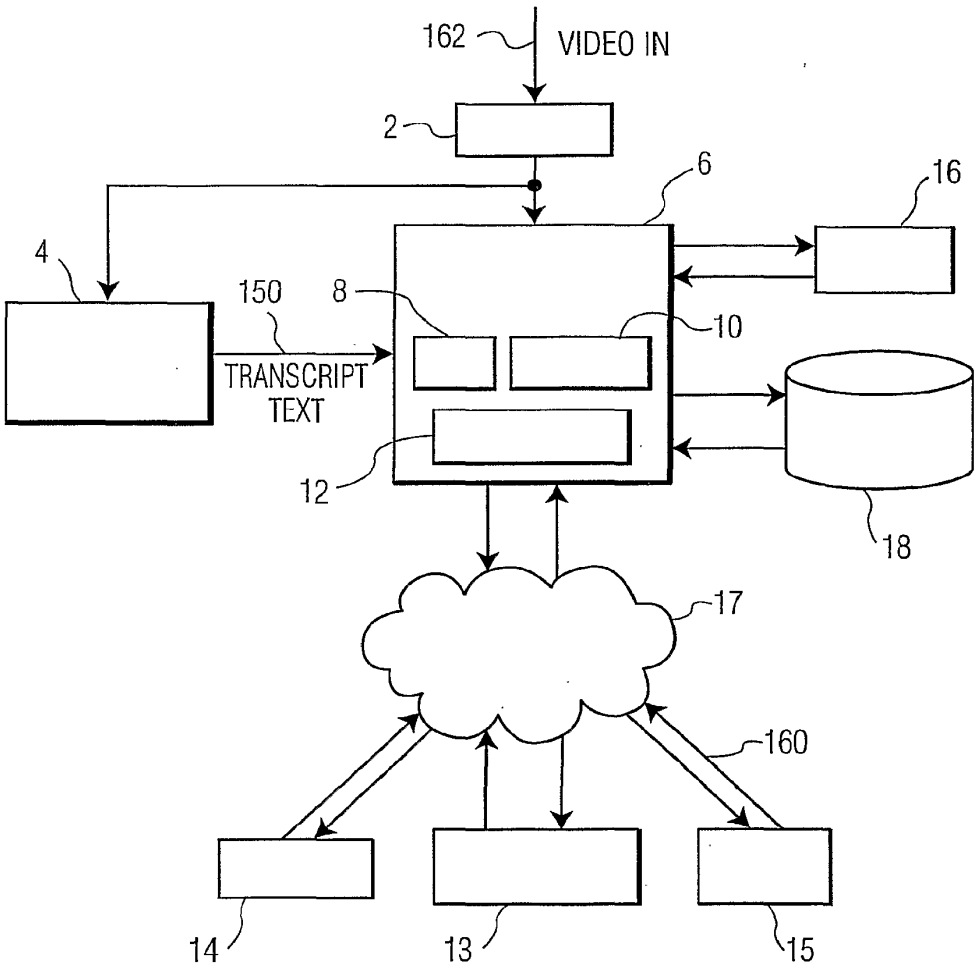

162
VIDEO IN
2
6
16
4
150
8
10
TRANSCRIPT
TEXT
12
18
17
160
14
13
15

FIG. 1

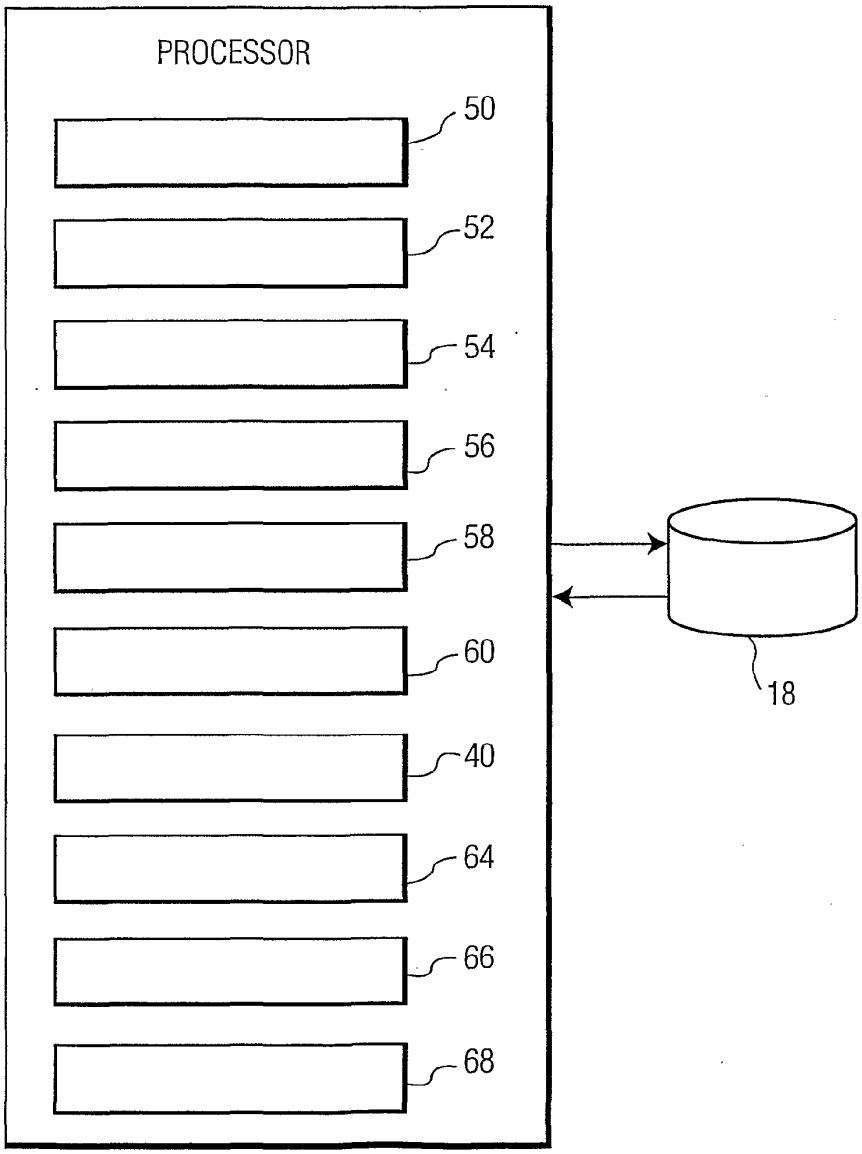


FIG. 2

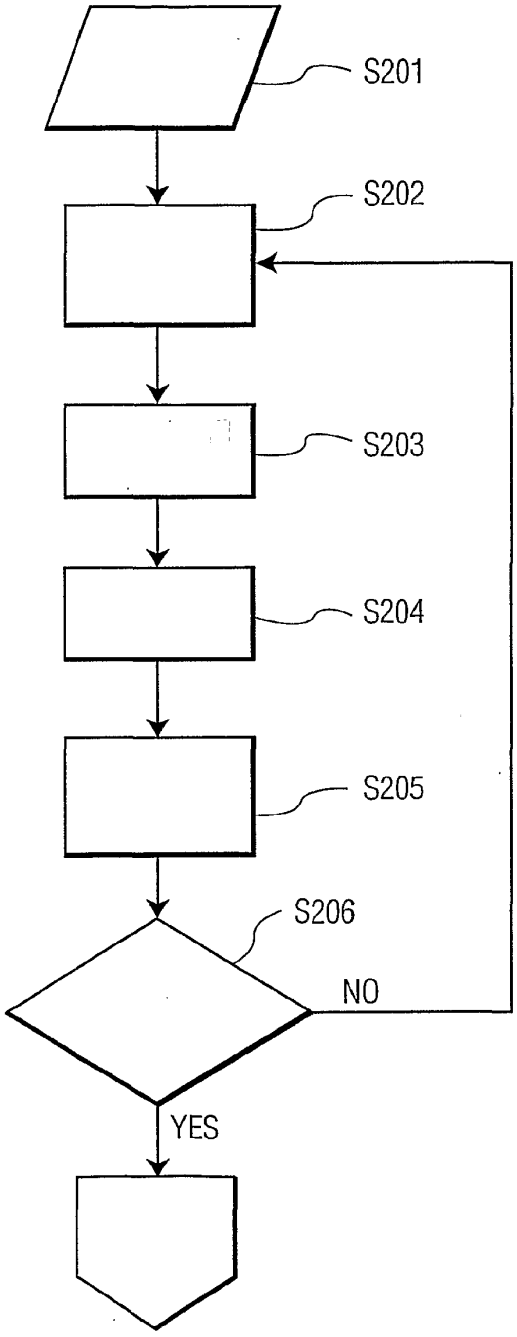


FIG. 3a

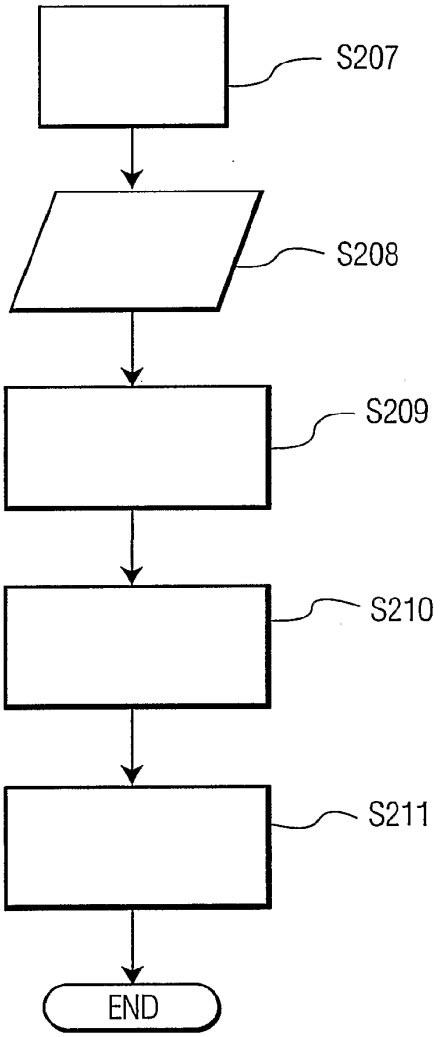


FIG. 3b

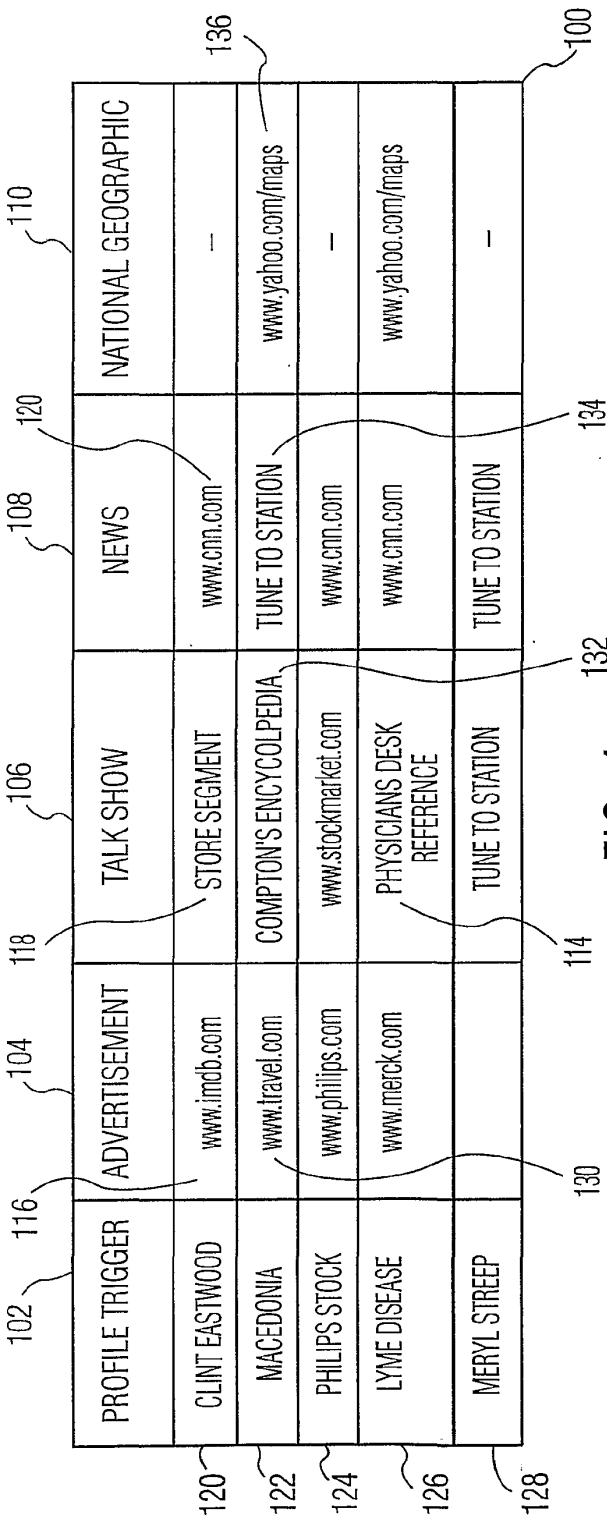

PROFILE TRIGGER
ADVERTISEMENT
TALK SHOW
NEWS
NATIONAL GEOGRAPHIC
CLINT EASTWOOD
www.imdb.com
STORE SEGMENT
www.cnn.com
—
MACEDONIA
www.travel.com
COMPTON'S ENCYCOLPEDIA
TUNE TO STATION
www.yahoo.com/maps
PHILIPS STOCK
www.philips.com
www.stockmarket.com
www.cnn.com
—
LYME DISEASE
www.merck.com
PHYSICIANS DESK REFERENCE
www.cnn.com
www.yahoo.com/maps
MERYL STREEP
TUNE TO STATION
TUNE TO STATION
—
102
104
106
108
110
116
118
120
120
122
124
126
128
130
132
134
136
100
114

FIG. 4

TRANSCRIPT TEXT 150

COMING UP NEXT IS A DISCUSSION OF LYME DISEASE WITH DR. JOHN JONES

KEYWORD 152

LYME DISEASE

KEYWORD EXPANSION 154

LYME DISEASE
TICK
TICK BITE
BULL'S EYE RASH
DEER TICK

TRIGGER 102

LYME DISEASE

TRIGGER EXPANSION 156

LYME DISEASE
TICK BITE
WEST NILE VIRUS
MOSQUITO SPRAYING

FIG. 4a

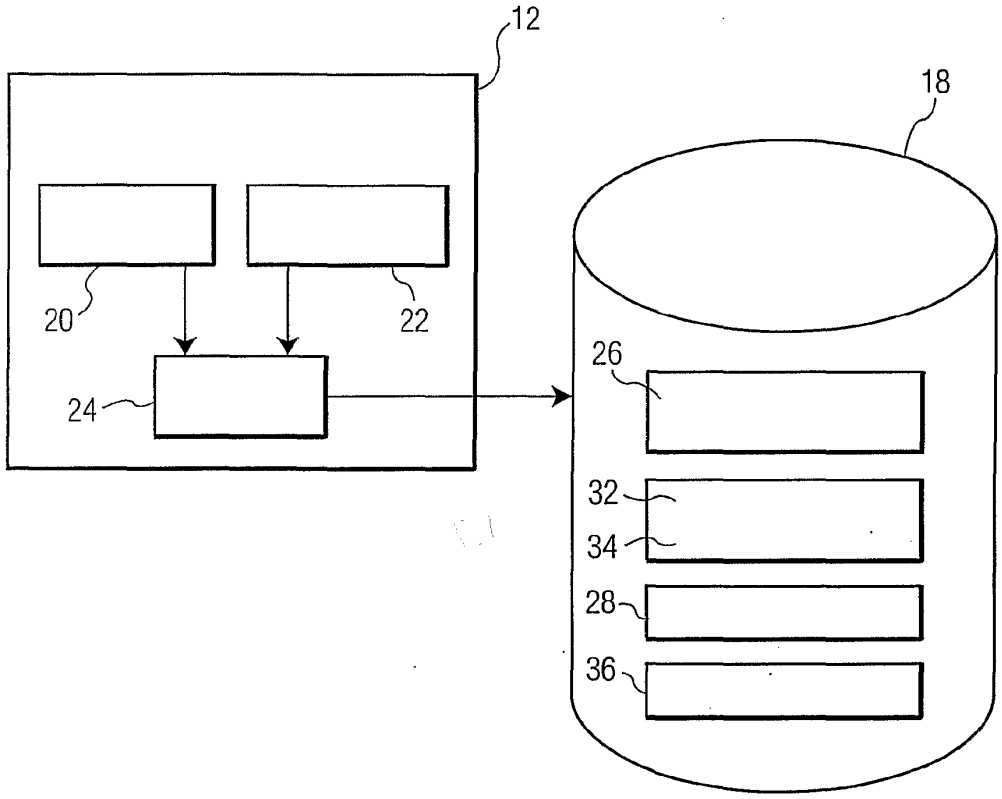


FIG. 5

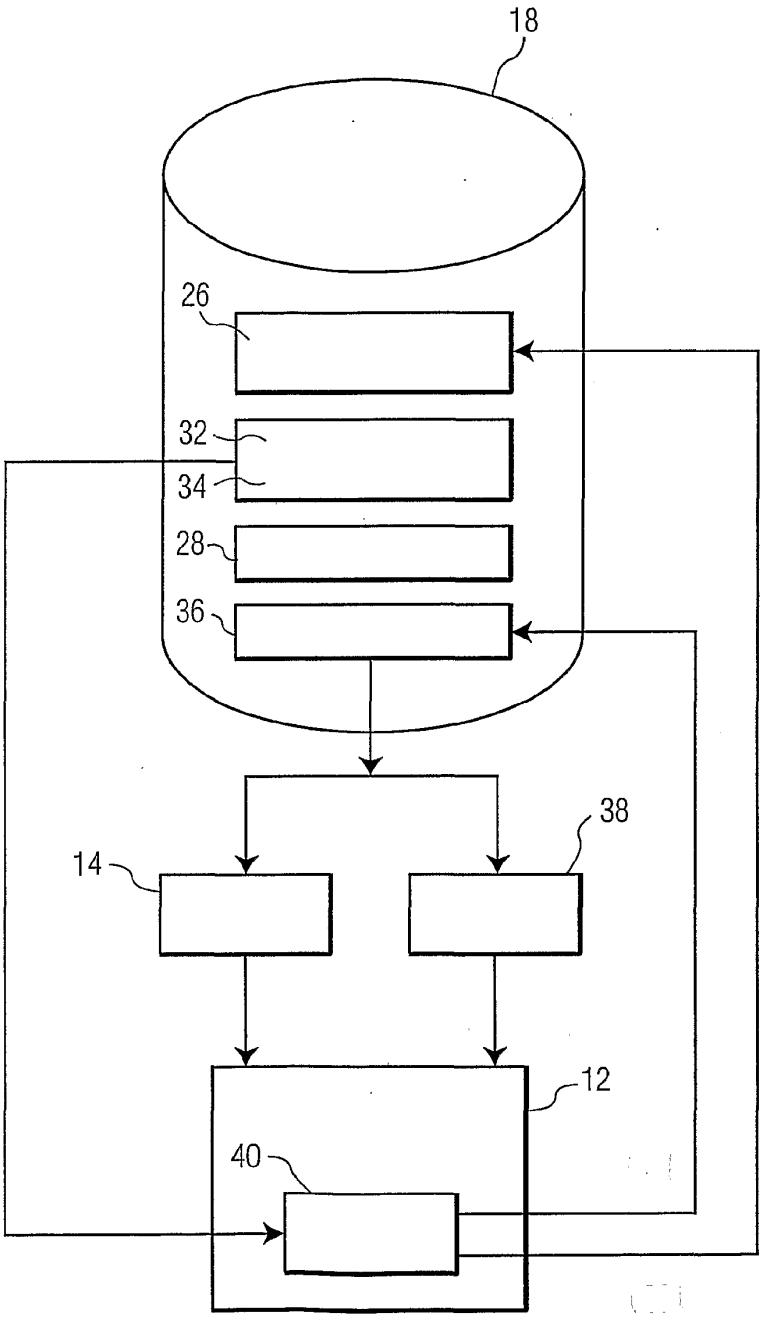


FIG. 6

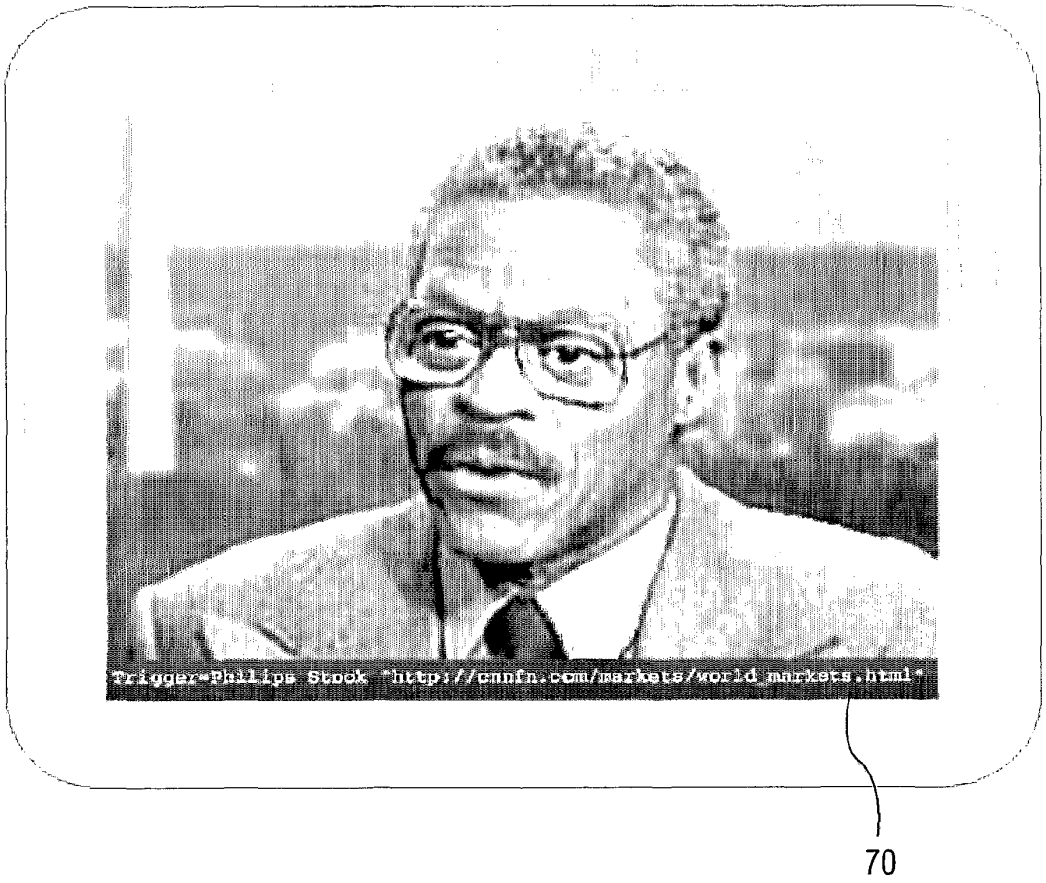

Trigger-Philips Stock "http://cnnfn.com/markets/world_markets.html"
70

FIG. 7

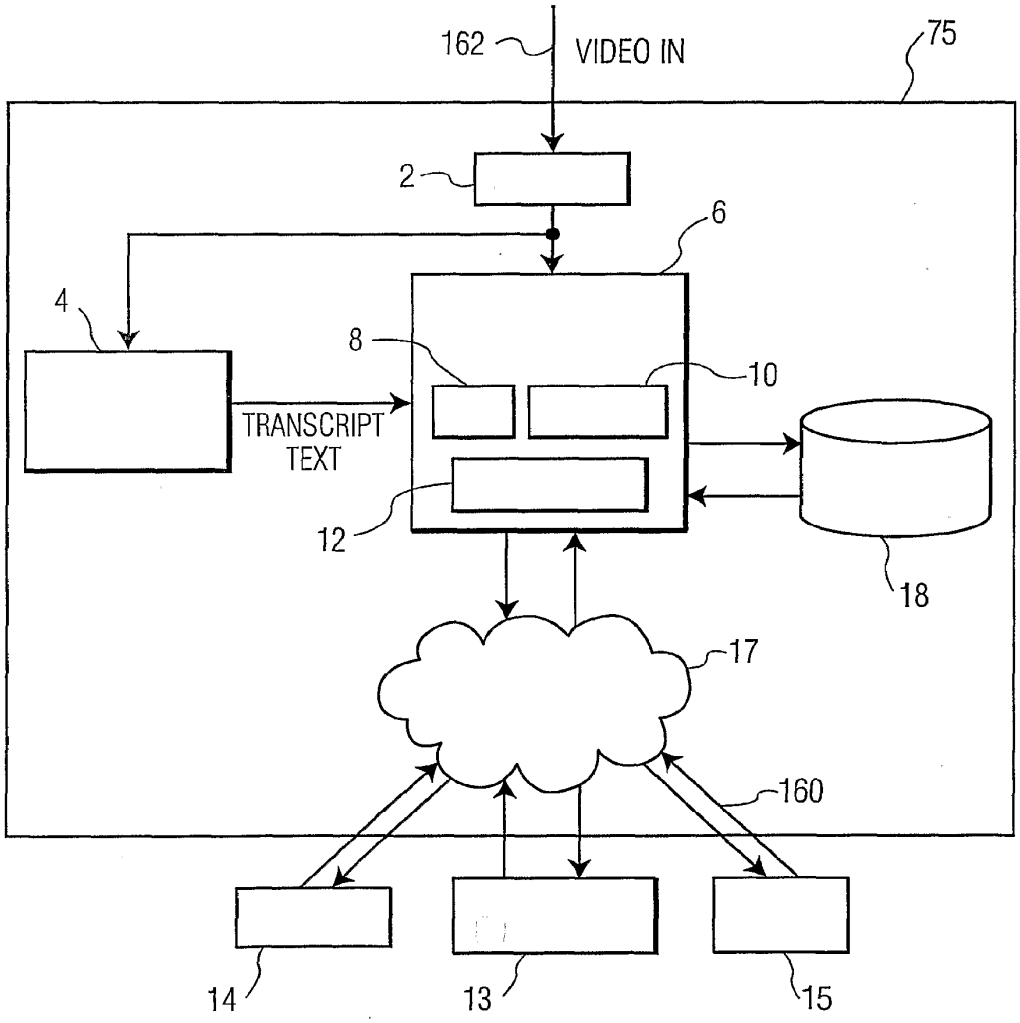


FIG. 8

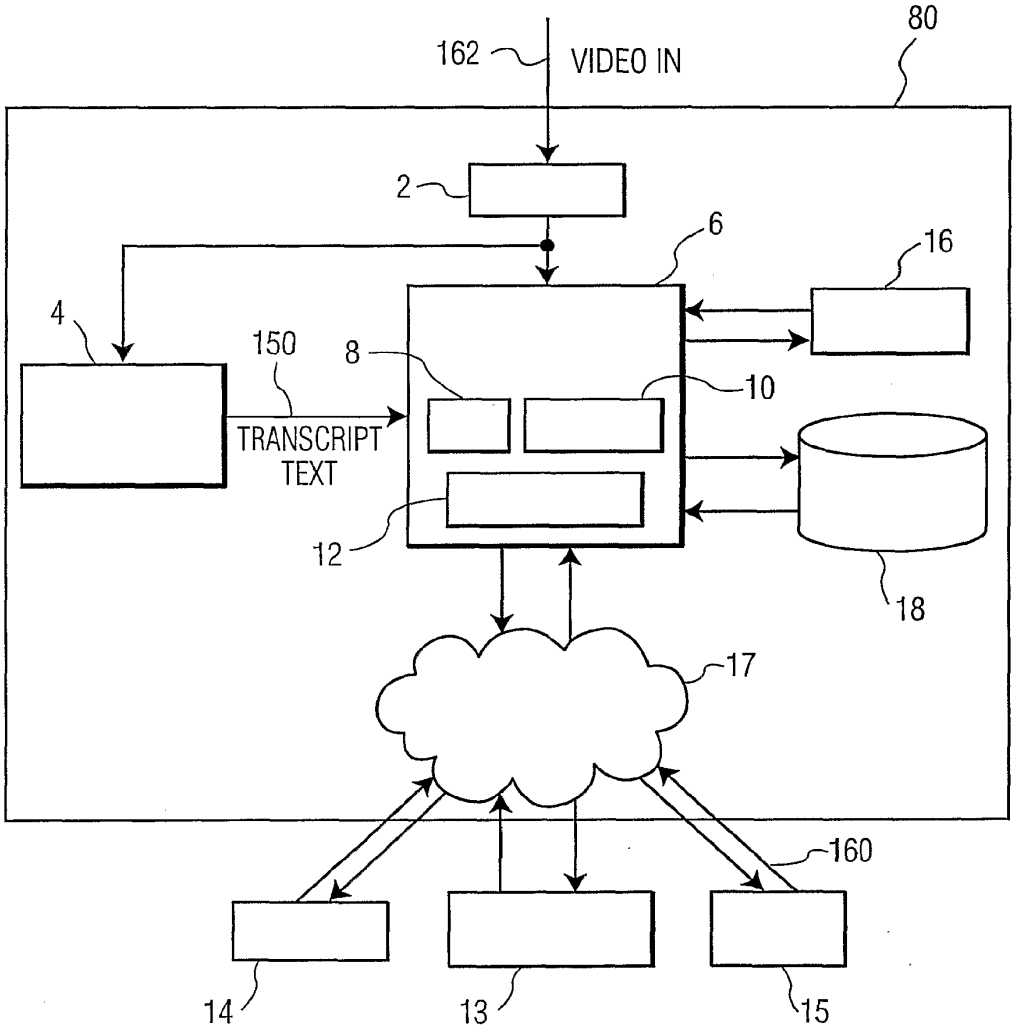

80
162
VIDEO IN
2
6
16
4
150
8
10
TRANSCRIPT
TEXT
12
18
17
160
14
13
15

FIG. 9

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06K 9/00	A1	(11) International Publication Number: WO 98/27510
		(43) International Publication Date: 25 June 1998 (25.06.98)

(21) International Application Number: PCT/US97/23291

(22) International Filing Date: 16 December 1997 (16.12.97)

(30) Priority Data:
08/772,188 19 December 1996 (19.12.96) US

(71) Applicant: THE REGENTS OF THE UNIVERSITY OF CALIFORNIA [US/US]; Business & Patent Law, Mail Stop D412, Los Alamos, NM 87545 (US).

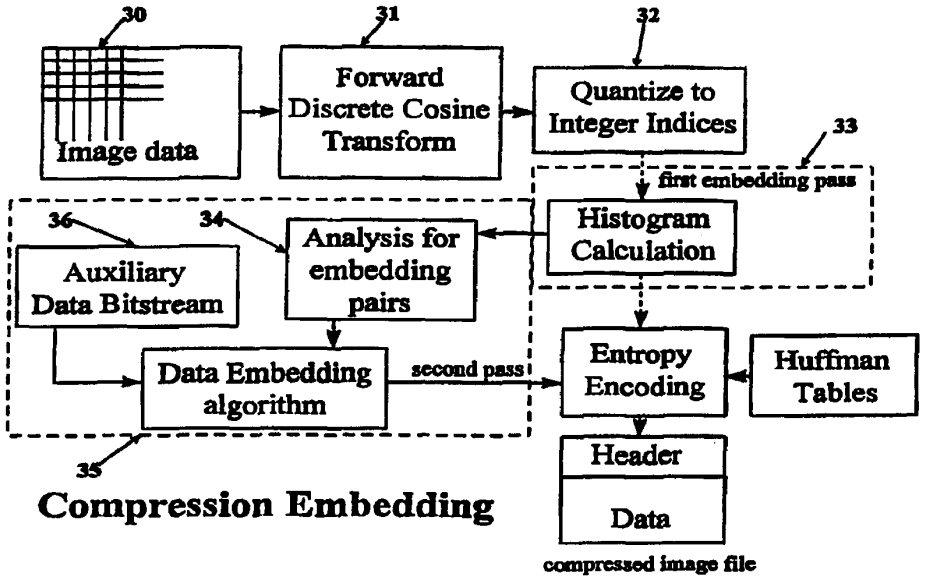
(71)(72) Applicants and Inventors: SANDFORD, Maxwell, T., II [US/US]; 160 Monte Rey Street, Los Alamos, NM 87544 (US). HANDEL, Theodore, G. [US/US]; 316 Bryce Avenue, Los Alamos, NM 87544 (US). BRADLEY, Jonathan, N. [US/US]; Apartment 512, 505 Oppenheimer Drive, Los Alamos, NM 87544 (US).

(74) Agents: WYRICK, Milton, D. et al.; Los Alamos National Laboratory, Mail Stop D412, Los Alamos, NM 87545 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: COMPRESSION EMBEDDING



(57) Abstract

A method and apparatus for embedding (35) auxiliary information into the digital representation of host data created by a lossy compression technique are presented. Also presented are a method and apparatus for constructing auxiliary data (36) from the correspondence between values in a digital key–pair table with integer index values existing in a representation of host data created by a lossy compression technique.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

COMPRESSION EMBEDDING

FIELD OF THE INVENTION

The present invention generally relates to digital manipulation of numerical data compressed by 'lossy' compression methods. More specifically, the invention relates to a means for the embedding of external data into the intermediate numerical values determined by the lossy compression method. This invention was made with Government support under Contract No. W-7405-ENG-36 awarded by the U.S. Department of Energy. The Government has certain rights in the invention.

BACKGROUND OF THE INVENTION

This is a continuation-in-part application out of U.S. Patent Application Serial Number 08/442,592, filed May 17, 1995.

The use of data in digital form is revolutionizing communication throughout the world. Much of this digital data communication is over wire, microwaves, and fiber optic media. When it is necessary to transmit large amounts of data, innovative methods to minimize the communication time are required. Transmissions of digital television, for example, use data compression methods to accomplish this minimization. A class of these methods is termed "lossy compression." The class is termed 'lossy' because the compression methods reduce slightly the quality of the original data. Multi-media computing applications use lossy compression of image and audio data to improve performance and reduce data storage requirements.

In many situations, it may be of benefit to send particular secondary data along with the primary data. Secondary data could involve the captioning of television programs, identification information associated with photographs and audio signals, data to authenticate or validate the host information, data impressed as a digital watermark signature, or command data, i.e., Java byte codes, transmitted with images in the Internet.

Authentication using a separate electronic indicia is taught in U. S. Pat. No. 4,949,381; to: Pastor; issued Aug. 14, 1990, assigned to Pitney Bowes, Inc., Stamford, Conn. This application teaches another method, wherein no separate indicia is required to convey the hidden information. The machine readable secondary, or auxiliary data reside entirely within the content of the host data. The auxiliary data are invisible to humans, and require a digital

key to be constructed from the host information by machines. The digital key may be protected by the use of "public key" encryption systems, such as the generally known RSA encryption system.

This application employs invisibility provided by manipulating the host information in a manner similar to the disclosures in copending application serial number 08/392,642, filed February 23, 1995, for DATA EMBEDDING. Auxiliary data embedded into the host information are constructed by processing machine readable, digital host data. Human readable versions of the host data, images displayed on screen or printed from the digital forms, cannot be scanned or processed to construct the auxiliary information. Data embedding as taught in the above copending application creates and manipulates a copy of the original image. The machine readable copy of the host image is manipulated to convey the added information. The amount of host information, i.e., the number of bits it contains is unchanged by data embedding, in contrast with authentication as taught by Pastor, who creates a separate, visible bit-mapped electronic indicia.

Authentication of the host data constitutes the successful construction of all or part of the auxiliary data embedded into the host information, because any modifications changing the host data representation preclude the invention.

In one embodiment of the subject invention, the auxiliary data are encrypted before the data embedding process in order to randomize the auxiliary bit stream and minimize the effect of embedding on the host data. However, the security of data embedded as taught in the above-mentioned copending application requires no decryption, owing to the noise-key used to control the embedding process.

In still another embodiment of the subject invention, the noise-key used to embed and construct the auxiliary data is encrypted using a public key encryption system such as RSA.

The auxiliary data can be embedded using methods to modify the noise component of the data as disclosed in the above-described copending application. However, lossy compression methods modify the noise component and degrade the fidelity of the original host data. This loss in fidelity destroys any information which has been embedded into the noise component of the host data, as is taught by the abovedescribed copending application. This means that lossy compression is incompatible with prior data embedding methods. The present invention implements the principles of data embedding to achieve the embedding of an

auxiliary data bit stream in host data compressed with lossy compression methods. The invention is particularly useful for embedding information required to remove a digital watermark from host data consisting of digital television or multi-media images compressed with lossy methods.

Redundancy and uncertainty are intrinsic to lossy compression methods. Two examples of lossy compression are the Joint Photographic Experts Group (JPEG) standard, and the Wavelet Scalar Quantization (WSQ) algorithm that has been adopted by the Federal Bureau of Investigation for the electronic interchange of digital fingerprint information. A similar compression standard established by the Moving Picture Experts Group (MPEG) is used for digital television and multi--media imagery. The JPEG and MPEG algorithms are based on the Discrete Cosine Transform (DCT) representation of the host data. The WSQ method is based on a representation of the host data in terms of wavelet functions. In the methods, the host data representation exists in an intermediate stage as a sequence of blocks of integer values referred to as 'indices.' At this intermediate stage, loss of fidelity occurs because the transform coefficients that represent the original data are quantized to a finite number of integer representations. The above mentioned JPEG, WSQ, and MPEG methods apply some form of loss-less compression to the integer coefficient blocks, resulting in doubly compressed data approximating the original image.

In the normal JPEG algorithm, the indices from the quantizer compress further with a loss-less entropy coder, usually a modified Huffman compression. An alternate method taught in U. S. Pat. No. 5,187,755; to: Aragaki, issued Feb. 16, 1993; assigned to Dainippon Screen Mfg. Co., Ltd., Kyoto, Japan represents part of the coefficient blocks by means of code tables. The code tables are established before the loss-less compression step is applied. Code tables represent parts of the coefficient blocks to within the error permitted by the standard deviation in the original pixel data. The index for a code table is used in place of parts of the coefficient matrix, thereby achieving an additional improvement in the data compression. The JPEG format is represented by ISO standards DIS 10918-1 and DIS 10918-2. The code table method is incompatible with the JPEG standards, and it therefore requires a special implementation, or a revised standard. The present application does not apply to lossy compression methods using code table representations for the coefficients in a block, because the invention manipulates the indices within a coefficient block.

Redundancy occurs in all lossy compression methods because the same integer values occur many times in the compression sequence of indices. Uncertainty occurs in the integer representation because the uncertainty in the original host data is present also in its transform representation. Indeed, the above mentioned code table compression method exploits redundancy in the integer coefficients and approximates parts of coefficient blocks with members of a plurality code table. The present invention depends upon the integer index values within a code block being individually uncertain by at least one unit of value.

Derek Upham implemented a manipulation of the quantizer parity as a means for embedding information into the JPEG algorithm (JSTEG source code May 25, 1993, available electronically at: ftp://ftp.csua.berkeley.edu/pub/cypherpunks/steganography/jsteg). The modified quantizer output is passed to loss less compression, and then to the compressed image file. Kineo Matsui and K. Tanaka ("Video-Steganography: How to Secretly Embed a Signature in a Picture," International Multimedia Assoc. Intell. Prop. Project. Proc., 1994) also manipulate the parity of the quantizer output, in synchronization with a bit pattern that is embedded as a repeating digital signature. Modifying parity to add information to embed information into the JPEG indices changes their statistical distribution. Modifying the quantizer parity disturbs the performance of the loss less compression process, resulting in a significant increase in the size of the compressed file size.

In the invention, the quantizer output flows to the data embedding algorithm, where key-pairs are calculated from constraints applied to the statistical distribution of the indices. Data bits from the auxiliary data modify the quantizer output according to the key-pair value correspondence, as taught in the above-described copending application. The advantage of the invention is two-fold:

1) The statistical properties of the quantizer output determine pairs of indices that can be exchanged without affecting significantly the entropy of the indices. Thus, the embedding process is nearly adiabatic, and the loss less coding performs nearly as well as it does without embedding data.

2) The histogram analysis leads naturally to a schedule of indices, or key-pair table, that drives the embedding algorithm. The security of the embedded information is high, because the embedding is done according to key-pair values that must be known to retrieve the information.

For embedding with the invention, the quantizer output flows to the data embedding algorithm, where the key-pairs and data bits from the auxiliary data are used to modify the quantizer output.

The existence of redundancy and uncertainty in the intermediate compression indices permits a process for embedding additional, invisible information into the machine readable compressed representation of the host data. Because the expansion of the compressed data returns an approximation to the original information, the adherence to the existing standards for JPEG and MPEG compression is unaffected by embedding the additional information. The embedded information is invisible to humans viewing the image expanded from the lossy compression data, and to machines reading the coefficient indices for purposes of reconstructing the host data. The auxiliary information is constructed by a machine that processes the indices and has the requisite knowledge to interpret certain of their values, in particular ones matching the key-pair schedule entries, according to their position of occurrence in the host data. The machine required to process the embedded information, for either embedding or auxiliary data construction, can be a channel in series with, or parallel to, the hardware or software implementing the compression expansion algorithm.

The embedding of information by the invention does not increase the bandwidth required for the transmission of the compressed data because the auxiliary data reside in the identical coefficient values representing the original data. The positions of the integer values within their respective blocks of coefficients are changed to embed the auxiliary data, but this manipulation is nearly adiabatic, with consequent minimal affect on the final, loss-less compression of the coefficient blocks. One may convey thereby meaningful, new uncorrelated information invisibly in the machine readable redundant coefficients of the lossy compression representation without it ever being detected by unauthorized persons.

This invention is based on the above-described copending application, but expands the data embedding principle of that application to a new regime. Compression embedding as taught in the present application is different than the data embedding as taught in the copending application because compression embedding adds additional data by manipulating the intermediate numerical representation of the host created by a lossy compression method instead of manipulating the host data values directly, as is taught in the copending application.

It is therefore an object of the present invention to provide apparatus and method for embedding data into a digital information stream so that the digital information is not changed significantly.

It is another object of the present invention to provide apparatus and method for thwarting unauthorized access to information embedded in data compressed with lossy algorithms.

It is yet another object of the present invention to provide apparatus and method for allowing construction of an auxiliary data bit stream from a lossy, compressed digital information stream into which the auxiliary data bit stream has been embedded according to the present invention.

Additional objects, advantages, and novel features of the invention will be set forth in part in the description which follows, and in part will become apparent to those skilled in the art upon examination of the following, or learned by practice of the invention. The objects and advantages of the invention may be realized and attained by means of the instrumentalities and combinations particularly pointed out in the appended claims.

SUMMARY OF THE INVENTION

In accordance with the purposes of the present invention there is provided a method of embedding auxiliary data into host data compressed by lossy compression ratio methods that use series expansion and quantization techniques comprising the steps of creating a duplicate digital compression representation of said host data in terms of intermediate integer indices representing coefficients, the digital compression representation having certain statistical properties; creating a digital representation of the auxiliary data in the form of a sequence of individual bit values; evaluating the digital compression representation to determine pairs of integer indices differing by a specified number of units and occurring with similar frequency in the digital compression representation; replacing individual integer indices in the digital compression representation with values taken from a digital key comprising a schedule of the pairs of integer indices in order to embed individual bit values corresponding to the sequence of individual bit values of the auxiliary data; and outputting the digital compression representation with the auxiliary data embedded into a file format specified for the compressed data.

In another aspect of the present invention, and in accordance with its objects and principles, a method of constructing embedded auxiliary data from a machine readable lossy digital compression representation containing integer indices comprises the steps of creating a look-up table defining a correlation between the integer indices and the elements of a key-pair table; matching the decoded integer indices from the lossy digital compression representation with entries in the look-up table, for the purpose of determining correspondence of the integer indices with elements in the key-pair; constructing the auxiliary data according to the correspondence in the look-up table between the integer indices and an element in the key-pair table; validating the content and correctness of the auxiliary data by examining header data constructed from the correspondence between the integer indices and the key-pair values; and interpreting the auxiliary bits as data unrelated to the lossy compressed data.

In yet another aspect of the present invention and in accordance with its objects and principles, apparatus for embedding auxiliary data into individual frames of host data having integer indices of certain values and frequencies of occurrence compressed by lossy compression ratio methods that use series expansion and quantization techniques comprises buffer storage means receiving the individual frames of host data for entropy decoding, modifying and momentarily storing the integer indices of the individual frames of host data,,entropy coding said modifications to the integer indices of the individual frames of host data, and outputting the entropy coded modifications to the integer indices of the individual frames of host data. Histogram analysis means receive the entropy coded modifications to the integer indices of the individual frames of host data for processing a statistical relationship between the frequency of occurrence and the value of the integer indices of the individual frames of host data, identifying embedding pair candidates in the individual frames of host data, and outputting the embedding pair candidates. Key-pair table means receive the embedding pair candidates for creating a key-pair table for each of the individual frames of host data, and outputting the embedding pair candidates as a randomized table. Compression embedding means receive the embedding key pair table, the integer indices from the individual frames of host data retained in the buffer storage, and the auxiliary data for embedding the auxiliary data and the key-pair table into the individual frames of host data and outputting an embedded stream of individual frames of host data.

In still another aspect of the present invention and in accordance with its objects and principles apparatus for constructing auxiliary data and a key-pair table embedded into individual frames of host data compressed by lossy compression ratio methods that use series expansion and quantization techniques comprises look-up table means receiving the key-pair table from the individual frames of host data for extracting the key-pair table and outputting a key-pair index corresponding to each of the integer indices from the individual frames of host data in a buffer storage. Auxiliary data construction means receive the key pair table for each of the individual frames of host data and the integer indices from the individual frames of host data stored in the buffer storage for using the key pair to construct the auxiliary data from the individual frames of host data and outputting an auxiliary data bit stream.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and form a part of the specification, illustrate the embodiments of the present invention and, together with the description, serve to explain the principles of the invention. In the drawings:

FIGURE 1 is a reproduction of the gray scale image example used to describe the invention.

FIGURE 2 is a plot of the histogram of the integer representation of the image in Figure 1 compressed with the JPEG method.

FIGURE 3 is a plot of the histogram of the integer representation of the image in Figure 1 compressed with the WSQ method.

FIGURE 4 is a block diagram illustrating the processes used in the embedding of data into the integer indices of a lossy-compressed data host.

FIGURE 5 is a block diagram illustrating the construction of auxiliary data from the key-pair values and the index values in the lossy compressed data host.

FIGURE 6 is a listing of computer code used for determining the pairs of integer indices suitable for embedding data.

FIGURE 7 is a listing of computer code used for eliminating duplicate host data pairs.

FIGURE 8 is a listing of computer code that performs a randomization of the pairs of indices in order to provide security against unauthorized access to the embedded information.

FIGURE 9 is a listing of computer code that performs the actual embedding of auxiliary data into the compression representation.

FIGURE 10 is a listing of computer code that constructs auxiliary data from the key-pair values and the lossy compression host data indices.

FIGURE 11 is a block diagram illustrating the embedding method of the present invention in concert with lossy compression hardware.

FIGURE 12 is a block diagram illustrating the construction of auxiliary data in hardware according to the present invention.

DETAILED DESCRIPTION

The present invention allows data to be embedded into information compressed with a lossy method without naturally discernible alteration of the content of the compressed information or the meaning of the embedded data constructed from the compressed information. This is made possible because of the technique of the present invention, in which similar integer indices in a lossy compression representation are re-ordered according to the desired embedded or implanted information. The lossy compression examples are digital images compressed using JPEG and WSQ methods. The resulting images contain embedded information without that information being readily discernible to an unauthorized observer .

The JPEG method is given in an article by Gregory K. Wallace, "The JPEG Still Picture Compression Standard," **Communications of the ACM**, April 1991 (vol. 34 no. 4), pp. 30-44. The JPEG algorithm is used primarily for compressing digital images. A PostScript file containing a revised version of the article is available for electronic reference at: ftp.uu.net, graphics/jpeg/wallace.ps.Z. A somewhat less technical, more leisurely introduction to JPEG can be found in **The Data Compression Book** by Mark Nelson, published by M&T Books (Redwood City, CA), 1991, ISBN 1-55851-216-0. A detailed textbook about JPEG is **JPEG Still Image Data Compression Standard** by William B. Pennebaker and Joan L. Mitchell, published by Van Nostrand Reinhold, 1993, ISBN 0-442-01272-1.

The Independent JPEG-Group's C-language source code is available electronically from: ftp.uu.net (Internet address 137.39.1.9 or 192.48.96.9). The most recent released version can always be found there in directory graphics/jpeg.

The WSQ method as applied to compressing digital fingerprint images was given by J. N. Bradley and C. M. Brislawn, "The wavelet/scalar quantization standard for digital fingerprint images," **Proc. of the 1994 IEEE Intern. Symp. on Circuits and Systems,** vol.

3, pp. 205-208, 1994, and by J. N. Bradley, C. M. Brislawn, and T. E. Hopper, "The FBI wavelet/scalar quantization standard for gray-scale fingerprint image compression," **Proc. SPIE,** vol. 1961, pp. 293-304, April, 1993. Documentation for WSQ compression is available through Jonathan N. Bradley, Los Alamos National Laboratory, P. O. Box 1663, MS-B265, Los Alamos, NM 87545, and electronically from the Internet FTP site "ftp.c3.lanl.gov" in directory /pub/WSQ.

The JPEG method processes the host image in pixel blocks specified to the algorithm at the time the indices are calculated. The WSQ method compresses the host image by passing it through a series of multirate filters. In both the JPEG and WSQ algorithms, the image host data exist in an intermediate stage as a sequence of integer (16-bit signed) indices. The indices represent an image originally presented in a standard digital format. The characteristic of lossy compression that makes possible the invention is redundancy and uncertainty. Each integer index occurs typically many times in the compressed representation, and each index is uncertain in value due to uncertainty in the host data. The present invention creates a histogram of the integer indices in the compressed representation. This histogram shows the probability density of the integer values in the representation. The histogram plots the number of times a particular value occurs versus the value. For JPEG compression, values in the range ±1024 are sufficient, and for WSQ compression values in the range ±4096 are considered. A particular histogram of index values will depend on the image content, but both compression methods concentrate many values in a pattern symmetrical about 0.

Figure 1 is a reproduction of the gray scale image example. The invention works equally well with color images. The gray scale image example in Figure 1 does not represent a limit to the application of the invention. The gray scale example image is reproduced in Figure 1 at reduced scale and quality to facilitate printing. The reproduction in Figure 1 demonstrates only the nature of the image example, and does not convey a representation of the image quality appropriate to the invention. Images expanded from compression representations with data embedded are visually identical to those expanded from unmodified compression data.

Histograms of the indices for the image example in Figure 1 are shown in Figures 2 and 3, for JPEG and WSQ compression, respectively. Figure 2 shows the histogram for the JPEG compression representation, for the image sample shown in Figure 1, and compression ratio

about 12:1. The file size for the JPEG version of Figure 1 is 42953 bytes. Analysis of this image according to the invention identifies 50 pairs of values in the histogram, totaling 3573 bytes of embedding space.

Figure 3 shows the histogram for the WSQ coefficient representation of the image sample shown in Figure 1, and compression ratio about 20:1. Analysis of these coefficients according to the invention identifies 37 pairs of values in the histogram, totaling 471 bytes of embedding space. The WSQ histogram contains fewer coefficients than produced by the JPEG method owing to the larger compression ratio, but the invention nevertheless permits embedding information into the compressed representation.

The actual embedding of the auxiliary data into the compressed representation of integer indices in the JPEG algorithm is a two-part process, the steps of which are shown in Figure 4. The steps of the invention are shown in the dashed-outline portion of Figure 4.

The JPEG algorithm divides the original, host image 30 into blocks of 8 x 8 pixels. The pixel values within a block are converted to coefficients in a discrete cosine transform 31, and the coefficient values converted to indices by a quantizer 32. In the first invention step 33, the quantizer indices, representing the transform coefficients, are examined to identify pairs of the integer indices having values that occur with approximately the same statistical frequency, and that differ in value by only one unit. A key-pair table 34 is constructed from the histogram of the quantizer indices. The order of the integer indices key-pair values is randomized to generate a unique key sequence that cannot be duplicated by an unauthorized person.

In the second invention step 35, the pairs of indices identified from the histogram analysis are used in a second pass through the image to re-order the quantizer indices in accordance with the correspondence between the key-pair values and the bit values in the sequence of auxiliary data bits 36. The key-pair sequence 35 is saved separately, as a small binary data file.

The method for constructing auxiliary data from a key-pair schedule and the indices in a lossy compressed data host is shown in Figure 5. The key-pair sequence 41 of pairs of index values is recovered from the compressed data file 40 as taught in the above-described copending application, or it is supplied as information separate from the compressed data. The key-pair table 41 specifies the pair-values of indices differing by one unit in value, and

the order in which indices are used to construct the auxiliary data. The indices of the pairs in the key-pair schedule are used to construct a look-up table 42. For every quantizer index value that is possible, the value of a corresponding key-pair index is given. Index values having no correspondence with a key-pair, i.e., quantizer index values that are not used for embedding are assigned the index value -1.

The lossy compressed data residing in the host file 40 are processed with the appropriate entropy decoder, usually the Huffman table algorithm 43, to recover the quantizer index values. The quantizer index value is used as an entry in the look-up table to determine the key-pair corresponding to the index value. If the look-up table entry is -1, there is no corresponding key-pair and the next value from the entropy decoder is examined. If the look-up table indicates a correspondence, the index value is compared with the corresponding key-pair values (44). The key-pair value matching indicates the auxiliary bit value to be added to the bit stream 45 being constructed. If the index value is identical to the first element of the key-pair, the auxiliary bit value is zero, and if the index value is identical to the second key-pair element, the auxiliary bit value is one.

As the quantizer index values flow from the Huffman decoder 43 and the look-up table is applied 44, the auxiliary data bit stream 45 is constructed. The key-pair data are used first to construct header information. The header information specifies the length and the file name of the auxiliary data, and serves to validate the key. If the compressed host file contains no embedded information, or if the incorrect key sequence is used, the header information cannot be interpreted correctly. However, successful interpretation of the header data guides the construction of the auxiliary data.

As shown in Figure 5, the invention constructs the auxiliary data 45 without requiring an expansion of the lossy compressed data to a host image, i.e., there is no particular correspondence of the values in the look-up table 42 with the pixel values in the decompressed image. If decompression to an image is required, the auxiliary data construction method can proceed as an optional, parallel process. However, if construction of the auxiliary data 45 alone is required, the invention proceeds as shown in Figure 5.

The principle of data embedding according to the present invention involves the rearrangement of certain intermediate, integer values in a lossy compression representation in order to encode the values of the auxiliary data which is to be added. For the purposes of this

description of the invention, consider the compression representation to consist of a sequence of integer values or indices. Further assume that any intermediate index value is uncertain by ±1 unit in value. The frequency of occurrence or histogram value of a certain index i is f(i).

Two values i and j in the table of indices are candidates as embedding pairs if:

$$|i - j| = 1. \qquad 10$$

Index values meeting the criterion of Equation 10, and occurring also in the representation with $f(i)-f(j) < \delta$, where f(i) and f(j) are the probability of occurrence of adjacent intermediate index values, and δ is the tolerance imposed for statistical equality, are candidates for embedding use. The values i and j meeting this constraint constitute a pair of index values p_k. There are $k=0,1,2,...N_p$ such pairs in the compression representation, giving a total number M of embedding bits:

$$M = \sum_i f(i) + \sum_j f(j). \qquad 20$$

The summations of i and j run over a limited range of the intermediate indices. In the example given here, for JPEG compression, the summation limits are specified at ±1024.

It is now helpful to refer to Figure 6, wherein a partial listing of computer code in the C-language is printed. On the first pass through the lossy compression algorithm, the histogram table is filled with the number of occurrences of each index value, in the range from **MINVAL** to zero, and from zero to **MINVAL** (positive). For the JPEG algorithm, the range is (-1024,+1024). In the histogram analysis, the indices falling within the range ± **HISTLIMIT** are excluded from analysis as embedding pair candidates, because the low frequency coefficients affect markedly the appearance of the reconstructed image. The routine **jhistgm()** in Figure 6 accepts three calling arguments (line 57 in Figure 6, page 2). The first argument is a pointer to the histogram table that was valued during the first pass through the lossy compression algorithm. The second two arguments are pointers to the table of key-pairs, and the number of pairs. The latter two arguments are filled in value by the routine.

The histogram constraint applied in the loop starting at line 66 determines the candidate pairs. A detailed discussion of this loop code is given by Sandford and Handel in "The Data Embedding Method," Los Alamos National Laboratory report LA-95-2246UR, in **Integration Issues in Large Commercial Media Delivery Systems**, Proc. SPIE, **2615**, 226, ed. A. G. Tescher and V. M. Bove, Jr. (Philadelphia, PA 23-26 Oct. , 1995). Briefly, the pair values must differ by no more than one unit in value, and they must occur with a difference in frequency less than the average frequency for the two index values. Figures 2 and 3 show the histogram data for sample JPEG and WSQ compression representations. The code in line 76 in Figure 6, page 2, compares the absolute difference of the frequency of occurrence f(i) and f(j) with the average value of the two occurrence frequencies. Pairs differing by less than the average value are accepted in this example.

This simple selection scheme prevents an artificially large modification to the indices in the lossy compression representation. For example, if f(i)=1000 and f(j)=300, the absolute difference is 700 and the average is 650. This pair would be *rejected* as an embedding candidate. However, if f(i)=1000 and f(j)=750, the absolute difference and the average are 250 and 875, respectively, and this pair would be *accepted.* This, or a similar scheme for selecting pairs for approximate equality of their component's frequency of occurrence minimizes the effect of embedding on the performance of the entropy coding algorithm. Additionally, it minimizes the perceptible differences in the image expanded from the compressed representation containing embedded data.

The pairs selected from the histogram are stored in the data structure array element **pair[j]** in lines 77,78,and 79 (Figure 6, page 2). In the example in Figure 6, the data structure permits **MAXPAIRS** pairs to be selected. The structure element **pair[j].count** contains the total number of occurrences of the (i,j) values in the histogram table. Line 90 (Figure 6, page 3) uses the routine **duplicate()** to remove duplications from the pair table. The return value is the total number of pairs, **no_pairs**. The loop starting at line 91 calculates **lsum**, the total number of bits that can be embedded into the compression indices. The variable M, in Equation 20 defines the calculation performed in the loop starting at line 91.

The total amount of embedding space is tested against the size of the auxiliary file plus the size of the header in line 105 (see D. Upham 1993 for details). If the total embedding bit

count is less than the amount needed to store the auxiliary file and its header, an error message is printed and the number of pairs value is set to zero.

If embedding space sufficient to hold the auxiliary data exists, then the number of pairs found by the analysis is *reduced* to the minimum number required to hold the auxiliary information. For Upham's (1993) method of quantizer parity modification, the index modifications cease after the auxiliary file is embedded. Thus, there is a strong potential for a transition in image quality at the point in the image where the pixel blocks cease to be modified. In the invention, the pair values chosen for embedding exist throughout the image, in nearly all of the coefficient blocks. The loop to reduce the number of embedding pairs begins at line 111 in Figure 6, page 3.

After trimming the number of pairs, a routine is called to scramble the pair values. This routine performs two functions. It randomizes the sequence of pair values and it randomizes the correspondence between the auxiliary bit values and the elements within a key-pair. In the implementation shown in Figure 6, the pair order is not important because the auxiliary data bits are embedded sequentially into whichever is the next index value matching *any* pair in the key. To implement the order of the pairs in the key into the embedding process, the auxiliary data must be partitioned into blocks of bits, with each block corresponding to a particular key-pair.

However, the order of the elements within a key-pair is randomized, and this affects the embedding process. For example, if the first key-pair has elemental values (4,5), then the occurrence of an index value of 4 is taken to signify a zero-bit in the auxiliary file, and the occurrence of a 5, a one-bit. For another pair, say (8,9), the randomization might reverse the interpretation. Thus the pair would contain the elemental values (9,8) and 9 would correspond to a zero-bit in the auxiliary data. It is insufficient to know only the key-pair values. One must know also the *order* in which the values are assigned to correspond to auxiliary data bits, and if the auxiliary data is embedded in blocks of bits, the sequence in which the key-pairs correspond to the blocks.

The embedding process ignores completely the contribution the index values make to the compression representation. In JPEG compression, the values represent the coefficients in a discrete cosine transform performed over pixels in a square block of the image data. Usually, 8x8 pixel blocks are used, but the details of the transform and the tiling of the image data are

irrelevant for the invention. In WSQ compression, the indices are determined by quantizing the discrete wavelet transform coefficients which are calculated by repeated applications of a multirate filter bank. Again, details of the wavelet calculations and the sampling size are ignored in the selection and use of the embedding pairs.

Depending on the details of the selection algorithm, the key-pairs found can include generally redundant values. The same index value i, is found perhaps in several different pair combinations. Because multiple pairs cannot contain the same index entry, due to each pair combination of index values having to be unique, it is necessary to eliminate some pairs.

Referring now to Figure 7, the code fragment listed therein illustrates the manner in which duplicate pairs are eliminated by a separate routine. First, the histogram of the image is used to calculate the total number of occurrences of key-pair structure elements, with the code in Figure 6. The key-pair selection algorithm given in Figure 6 results in pairs being selected in ascending index order. Before rejecting some pairs, the pairs are sorted in line 23 of Figure 7, page 1 by order of decreasing frequency of occurrence value, **pair[j].count.** Sorting the pairs by occurrence frequency before rejecting duplicate pairs minimizes the number of pairs needed to embed a particular auxiliary data bit stream. The nested loops beginning at line 24 of Figure 7, page 1 reject duplicate pair values containing the fewest occurrence frequencies. The routine revises the pair structure elements in the loop starting at line 47 (Figure 7, page 2) and returns the new ***no_pairs** value.

The security of the embedded data is increased significantly if the pair values are arranged into random order. Randomizing the order of the pair values is part of this invention and follows the code listing in Figure 8. Randomizing is accomplished by rearranging the pair values according to a randomly ordered data structure. Two routines serve to randomize the key-pair values. The routine **scramble()** starts at line 20 in Figure 8, page 1, and the routine **jumble()** starts at line 74 in Figure 8, page 2. An ordered sequence of integers in the array named **index[]** is randomized by the **jumble()** routine. The random indices are used to re-order the key-pairs in a loop starting at line 34. The re-ordered data structure replaces the original key-pair data in line 39. In another loop starting at line 46 in Figure 8, page 2, the correspondence of the pair elements with the auxiliary bit values is randomized.

In the **jumble()** routine, the structure named **pts** contains elements **pts[k].i**, k=0,1,2,..., **no_pairs**; and **pts[k].gamma**, $\gamma_1,\ldots,\gamma_k,\ldots\gamma_{no\_pairs}$, where the γ_k values are uniformly random on

(0,1). The structure values are assigned in the loop starting at line 82 in Figure 8, page 2. The standard library routine **qsort()** is used to sort the data structure **pts[]**. Putting the random element values into ascending order randomizes the index element of the structure. The random index values are used with the pair structure elements calculated and sorted as indicated above, to re-order the table to give random pair ordering.

The actual embedding of auxiliary data into a compression representation consists of rearranging the order of occurrence of the redundant indices. A fast embedding implementation results by using a table look-up in place of the exhaustive search that was given previously in the above mentioned work by Sandford and Handel (1995). Figure 9 contains the code for the **Embed_Data_Block()** routine. The routine includes several static variables (Figure 9, lines 38-40, page 1; lines 44-49, page 2). The variable **ptable[]** is an array that contains the index of the pair that corresponds to any value of the quantizer output, also an index.

The arguments to the routine entry at line 55 in Figure 8, page 2, are a pointer to a buffer containing quantizer index values and the number of values that are to be processed. To find the embedding pair structure that corresponds to any quantizer output value, the value is offset by the minimum amount and used as an index into the **ptable[]** array. The value taken from **ptable[]** is the index to the correct pair of embedding values. Values in the **ptable[]** set to -1 indicate there is no correspondence with embedding key-pair values. Thus, quantizer values giving negative **ptable[]** values are ignored, and those giving **ptable[]** values greater than, or equal to zero, are used to embed one bit of auxiliary data. A static variable named **ptable_filled** is used to indicate that the **ptable[]** array must be initialized from the key. The test at line 61 in Figure 9, page 2 permits reading the binary key-pair file and setting the **ptable[]** values.

Data embedding is accomplished by the loop starting at line 97 in Figure 9, page 3. For each quantizer index value in the input buffer, the data embedding algorithm (Sandford and Handel 1995) is applied. The routine **bitgetbit()** (see Upham's 1993 JSTEG code for details) supplies the value of the next bit in the header or the auxiliary data file. Upon exhaustion of the auxiliary bits, the return value (line 103) is -1, and the embedding is completed. The commented line 107 is activated if it is desired to generate a 'marked' output image that shows the locations in the host data that are used by the key-pair values.

The key-pairs selected for embedding contain the index values to be used in the rearrangement. It is important to realize that the numerical values used for embedding data are the index values already occurring in the compression representation. The maximum pair table size permitted depends upon the compression representation. For JPEG compression, Figure 2 shows the index values concentrate near the origin and 50 to 100 pair values are adequate. For WSQ compression of digital fingerprints, the index values spread over a larger spectrum and more pairs are required.

A consequence of the histogram constraints used to determine the key-pairs is that the current invention alters the entropy in the DCT or WSQ coefficients only slightly. Thus, the efficiency of loss less compression is largely unaffected by embedding auxiliary data into the indices values.

In the embedding process of the present invention, the coefficients calculated by the compression algorithm are manipulated in the order they appear from the compression scheme used to generate the representation, JPEG and WSQ for the examples herein. The embedding process flows concurrently through the sequence of auxiliary data bits and the compression indices. Upon identifying a compression index matching one of the pair table values, the next bit in the sequence of auxiliary data is examined to determine if the index is set to the **pair[k].i** value (embedding a 0), or set to the **pair[k].j** value (embedding a 1). The key-pair table is processed sequentially, in the order found after it was randomized by the code in Figure 8.

The standard "Lena" test image is used to compare the results of the invention with JPEG baseline compression and quantizer parity steganography as implemented by Upham (1993). A 1288-byte test file (jembed.obj) is used as the auxiliary data, and the identical header procedure is implemented for the comparison.

Table 1 gives the file sizes for the original Lena image (lena_rgb.tga), for JPEG baseline compression of the original 24-bit Truecolor image (lena.jpg), for quantizer parity steganography with the JSTEG method (lenau.jpg), and for the invention (lenax.jpg). The Truecolor image compresses to the baseline JPEG standard with ratio 18.88:1 from the original size. With 1288 bytes embedded (3 % of the compressed size), quantizer parity steganography increases the compressed file size by 801 bytes (1.9 %), giving 18.52:1 compression ratio. The invention increases the file size 16 bytes (0.038 %) giving 18.87:1

compression ratio. Quantizer parity steganography increases the compressed file size by 62 % of the auxiliary file size. The present invention increases the compressed file size by 1.2 % of the auxiliary file size. Thus, the Lena test image demonstrates that the present invention affects minimally the compression efficiency of the JPEG algorithm.

Table 1. Test Images for 1288 bytes

file name	file size (bytes)
lena_rgb.tga	788,019
lena.jpg	41,747
lenau.jpg	42,548
lenax.jpg	41,763

The construction of auxiliary data from a lossy compressed host containing embedded data is achieved by processing the entropy decoded quantizer coefficients with a look-up table. Figure 10 gives the listing of code to construct auxiliary data from the quantizer indices. The routine named **ExtractDataBlock()** processes a buffer of quantizer indices and constructs the auxiliary bit stream associated with the key-pair values. The routine uses static variables (see lines 19 to 23 in Figure 10, page 1) for the look-up table **ptable[]**, a flag named **ptable_filled** to signify that the **ptable[]** array requires filling from the key-pair schedule, and a variable named **embed_count** to count of the number of auxiliary bits constructed. The subroutine begins at line 43 in Figure 10, page 2, and it accepts two arguments. The first argument is a pointer to the buffer containing integer values of the quantizer indices, and the second argument specifies the number of values in the buffer.

If the **ptable[]** look-up array requires filling, the key-pair schedule is read from a binary data file at line 57 in Figure 10, page 2, and the look-up table is filled with indices indicating the correspondence between the key-pair values and the quantizer indices. The loop starting at line 81 in Figure 10, page 2 processes the quantizer indices in the input buffer. If a correspondence between a quantizer index value and an entry in the key-pair table exists, the quantizer index is tested against the key-pair value elements to determine if the auxiliary bit is zero or one. The routine **bitsetbit()** is used to add one bit of the appropriate type to the auxiliary bit stream being constructed (see Upham 1993 for details). The **bitsetbit()** routine

returns -1 when the auxiliary data file is completely constructed. The return value is tested at line 88 in Figure 10, page 3.

Owing to the uncertainty introduced by randomizing the order of the key-pair table, the auxiliary data are secure against unauthorized construction from the compression representation. Indeed, detecting the presence of embedded data is difficult, if not impossible, because the only metric known to us for such detection is a statistical analysis of the compression representation indices. Because the invention affects the statistical properties only slightly, it leaves no characteristic signature revealing the presence of embedded information.

As previously described, the ordered pairs of index values selected for embedding constitute the "key" for construction of the auxiliary data from the compressed image. The listings illustrated in Figures 6 through 10 demonstrate how embedding analysis exploits redundancy and the statistical uncertainty in the compression representation to construct a look-up table of key-pairs of index values. The key-pairs are required for constructing the auxiliary data, but they cannot be generated by analyzing the compressed image representation after the embedding process is completed. The invention retains the one-time-pad encryption properties inherent in data embedding as disclosed in the above-described copending application, and incorporates the same high security to the embedded bit stream.

As for data embedding as disclosed in the copending application, another way of protecting the pair table key is to encrypt it using public-key or another encryption process. The present invention permits an encrypted key to be placed into the compressed file, preventing extraction by unauthorized persons.

Embedding auxiliary data into the compression representation slightly changes the statistical frequency of occurrence of the index values. If the auxiliary bit sequence is pseudo-random, the frequencies of occurrence for the index pairs i and j are nearly equal after embedding. Equalizing the histogram reduces entropy somewhat, and the entropy coding portion of the compression algorithm is found often to operate with slightly *greater* efficiency, increasing the effectiveness of the lossy compression method. Thus, even if no auxiliary information is to be embedded, equalizing the statistical properties of the histogram pairs improves the compression ratio for lossy methods. Embedding pseudo-random, or white noise to improve compression is part of this invention.

The present invention can be implemented into existing lossy compression machines, as shown in Figure 11. In the upper left of Figure 11, a standard, composite video signal 11 is input to encoder chipset 12 which encodes composite video signal 11. The binary output 13 of encoder chipset 12 consists of sequences of Inter-frames (I-frames), motion vector information, and Backward- and Forward-predicted frames (B,P-frames) and, when auxiliary information is to be embedded, passes via the dashed path to buffer 15 of embedding chipset 14.

Embedding chipset 14 consists of the functions within the dashed box in Figure 11. Embedding chipset 14 provides the functions which implement the present invention. As illustrated, buffer 15 entropy decodes, modifies and momentarily stores the integer indices of the individual frames of host data in preparation for the subsequent embedding procedure. The modification of the integer indices is conducted as described hereinabove in this specification, after which buffer 15 entropy codes the modifications to the integer indices. These modified integer indices are then output to histogram analysis 16, where the statistical relationship between the frequency of occurrence and value of the modified integer indices is processed in order to identify appropriate embedding pair candidates. These embedding pair candidates are output to key-pair table 17. Key-pair table 17 processes the embedding pair candidates to generate a key-pair table for a frame of binary output 13 which is stored in buffer 15.

Compression embedding engine 18 receives each frame of buffered binary output 15 as well as the output of key-pair table 17 and auxiliary bit stream 19. Compression embedding engine 18 embeds auxiliary bit stream 19 into the I-frame indices of buffered binary output 15, and the modified I-frame indices are compressed along with the key-pairs to produce embedded MPEG-2 bit stream 20

The processing time for embedding auxiliary data with the present invention is less than the time required to encode an I-frame and its associated motion-predicted frames (B- and P-frames), thereby maintaining the output rate of the MPEG-2 data stream. The key-pair table for the I-frame is inserted into the motion prediction portion of the MPEG-2 data, or it is added as a fictitious frame. Thus, the key-table data appear as an MPEG-2 coding error that is ignored by MPEG-2 decoding machines.

It should be understood that, although Figure 11 illustrates chipset 14 for implementing the present invention, the functions of chipset 14 could also be implemented in circuit board circuitry. Such implementations could be useful for limited applications of the present invention.

In Figure 12, apparatus for constructing auxiliary data from an embedded MPEG-2 bit stream is illustrated. The embedded MPEG-2 signal 21 is processed in parallel with a conventional MPEG-2 decoder 22 that generates composite NTSC standard video 23 for viewing. The present invention, in the parallel leg within the dashed box in Figure 12, obtains the key-pair table from the MPEG-2 entropy-decoded bit stream data and generates a look-up table 24 from the key-pair table to permit processing the embedded I-frame integer indices by auxiliary data construction 25. Auxiliary data construction 25 outputs auxiliary data stream 26.

Depending upon its content, auxiliary data stream 26, so constructed, can be used to correct or augment the NTSC decoded signal 23, or data stream 26 can be used for any other predetermined purpose. As an alternative to the stand-alone chipset or circuit board manifestation of the present invention illustrated within the dashed box in Figure 12, the invention can be incorporated into MPEG-2 decoder 22, in order that auxiliary data stream 26 is available to correct or augment the MPEG-2 bit stream fields.

The foregoing description of the preferred embodiment of the invention have been presented for purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed, and obviously many modifications and variations are possible in light of the above teaching. The embodiments were chosen and described in order to best explain the principles of the invention and its practical application to thereby enable others skilled in the art to best utilize the invention in various embodiments and with various modifications as are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the claims appended hereto.

WHAT IS CLAIMED IS:

1. A method of embedding auxiliary data into host data compressed by lossy compression ratio methods that use series expansion and quantization techniques comprising the steps of:

creating a duplicate digital compression representation of said host data in terms of intermediate integer indices representing coefficients, said digital compression representation having certain statistical properties;

creating a digital representation of said auxiliary data in the form of a sequence of individual bit values;

evaluating said digital compression representation to determine pairs of integer indices differing by a specified number of units and occurring with similar frequency in said digital compression representation;

replacing individual integer indices in said digital compression representation with values taken from a digital key comprising a schedule of said pairs of integer indices in order to embed individual bit values corresponding to said sequence of individual bit values of said auxiliary data; and

outputting said digital compression representation with said auxiliary data embedded into a file format specified for said compressed data.

2. The method as described in Claim 1 further comprising the step of combining said auxiliary data with predetermined information indicative of the presence of said auxiliary data, its file name, and file size, said step to be performed after the step of digitizing said auxiliary data.

3. The method as described in Claim 1 further comprising the step of equalizing said statistical properties of said digital compression representation for the purpose of improving said lossy compression ratio methods.

4. The method as described in Claim 1, wherein said digital compression representation originates from a color image compressed with the JPEG or WSQ algorithm, and their derivative compression representations.

5. The method as described in Claim 1, wherein said digital compression representation originates from a black and white image compressed with the JPEG or WSQ algorithm, and their derivative compression representations.

6. The method as described in Claim 1, wherein said digital compression representation originates from a television or video disc signal compressed with the MPEG and DCT-based algorithms, and their derivative compression representations.

7. The method as described in Claim 1, wherein said digital compression representation originates from digital audio represented by lossy compression representations.

8. The method as described in Claim 1, wherein said digital compression representation originates from X-ray, CT-scan, or Magnetic Resonance Imaging data.

9. The method as described in Claim 1, wherein said digital compression representation originates from personal security and identification information.

10. The method as described in Claim 1, wherein said digital compression representation originates from fingerprint images.

11. The method as described in Claim 1, wherein said digital compression representation originates from images made with a camera producing digital images.

12. A method of constructing embedded a quantity of auxiliary data from a machine readable lossy digital compression representation containing integer indices comprising the steps of:

creating a look-up table defining a correlation between decoded integer indices from said lossy digital compression representation and elements of a key-pair table;

matching said decoded integer indices from said lossy digital compression representation with entries in said look-up table, for determining correspondence of said integer indices from said lossy digital compression representation with said elements in said key-pair;

constructing said auxiliary data according to said correspondence in said look-up table between said integer indices from said lossy digital compression representation and an element in said key-pair table; validating the content and correctness of said auxiliary data by examining header data constructed from the correspondence between said integer indices and said key-pair values; and

interpreting said auxiliary bits as data unrelated to said lossy compressed data.

13. The method as described in Claim 12, wherein said auxiliary data consist of digital text.

14. The method as described in Claim 12, wherein said auxiliary data consist of script or code intended for execution on a network client or server machine.

15. The method as described in Claim 12, wherein said auxiliary data consist of motion vector data or script used in the reconstruction of a digital motion picture.

16. The method as described in Claim 12, wherein said auxiliary data consist of text or digital audio data used for subtitles or language translation of digital motion picture audio.

17. The method as described in Claim 12, wherein said auxiliary data consist of information used to correct degraded, or to decode scrambled image data, for the purpose of controlled service delivery.

18. The method as described in Claim 12, wherein said auxiliary data consist of additional or supplemental digital imagery.

19. The method as described in Claim 12, wherein said auxiliary data consist of personal identification information.

20. The method as described in Claim 12, wherein said auxiliary data consist of medical record or diagnostic information.

21. The method as described in Claim 12, wherein said auxiliary data consist of a portion of the said original lossy compressed data.

22. The method as described in Claim 12, wherein said auxiliary data consist of encrypted information.

23. The method as described in Claim 12, wherein said auxiliary data consist of information transmitted in the space of said lossy compressed representation afforded by the existence of a correspondence between said integer indices and said pair key table.

24. Apparatus for embedding auxiliary data into individual frames of host data having integer indices of certain values and frequencies of occurrence compressed by lossy compression ratio methods that use series expansion and quantization techniques comprising:

buffer storage means receiving said individual frames of host data for entropy decoding, modifying and momentarily storing said integer indices of said individual frames of host data ,entropy coding said modifications to said integer indices of said individual frames of host data, and outputting said entropy coded modifications to said integer indices of said individual frames of host data;

histogram analysis means receiving said entropy decoded integer indices of said individual frames of host data for processing a statistical relationship between said frequency of occurrence and said value of said integer indices of said individual frame of

host data identify embedding pair candidates in said individual frames of host data, and outputting said embedding pair candidates;

key-pair table means receiving said embedding pair candidates for creating a key-pair table for each of said individual frames of host data, and outputting said embedding pair candidates as a randomized table;

compression embedding means receiving said embedding key pair table, said integer indices from said individual frames of host data retained in said buffer storage, and said auxiliary data for embedding said auxiliary data and said key-pair table into said individual frames of host data and outputting an embedded stream of individual frames of host data.

25. The apparatus as described in Claim 24 wherein said buffer storage means, said key-pair table means, and said compression embedding means are contained within a chipset.

26. The apparatus as described in Claim 24 wherein said buffer storage means, said key-pair table means, and said compression embedding means are implemented in circuit board circuitry.

27 Apparatus for constructing auxiliary data and a key-pair table embedded into individual frames of host data compressed by lossy compression ratio methods that use series expansion and quantization techniques comprising:

look-up table means receiving said key-pair table from said individual frames of host data for extracting said key-pair table and outputting a key-pair index corresponding to each of said integer indices from said individual frames of host data in a buffer storage; and

auxiliary data construction means receiving said key pair table for each of said individual frames of host data and said integer indices from said individual frames of host data stored in said buffer storage for using said key pair to construct said auxiliary data from said individual frames of host data and outputting an auxiliary data bit stream.

28. The apparatus as described in Claim 27 wherein said look-up table means and said auxiliary data construction means are contained within a chipset.

29. The apparatus as described in Claim 27 wherein said look-up table means and said auxiliary data construction means are implemented in circuit board circuitry.

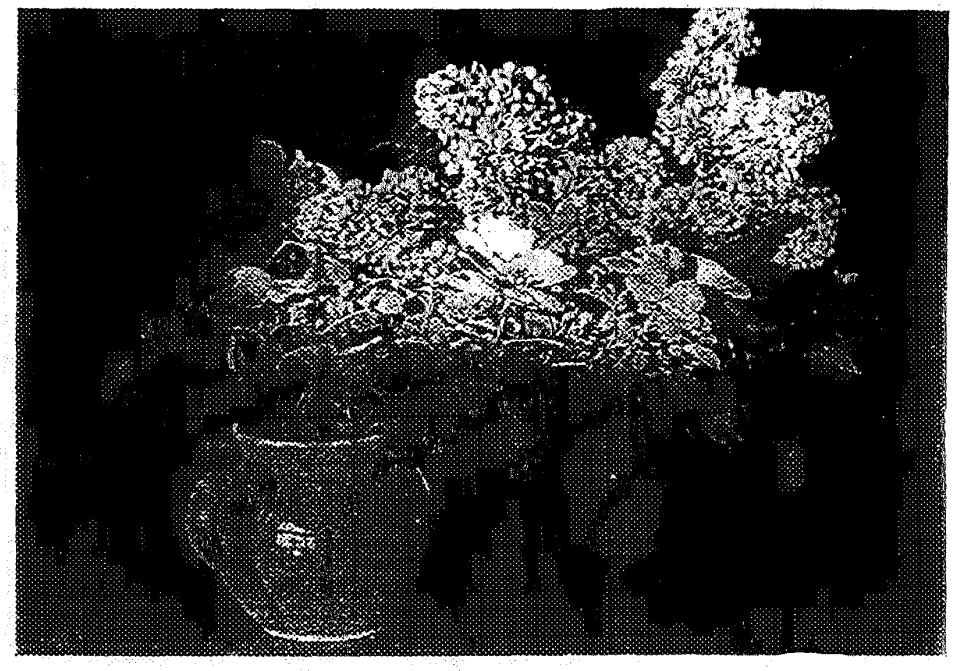


FIGURE 1

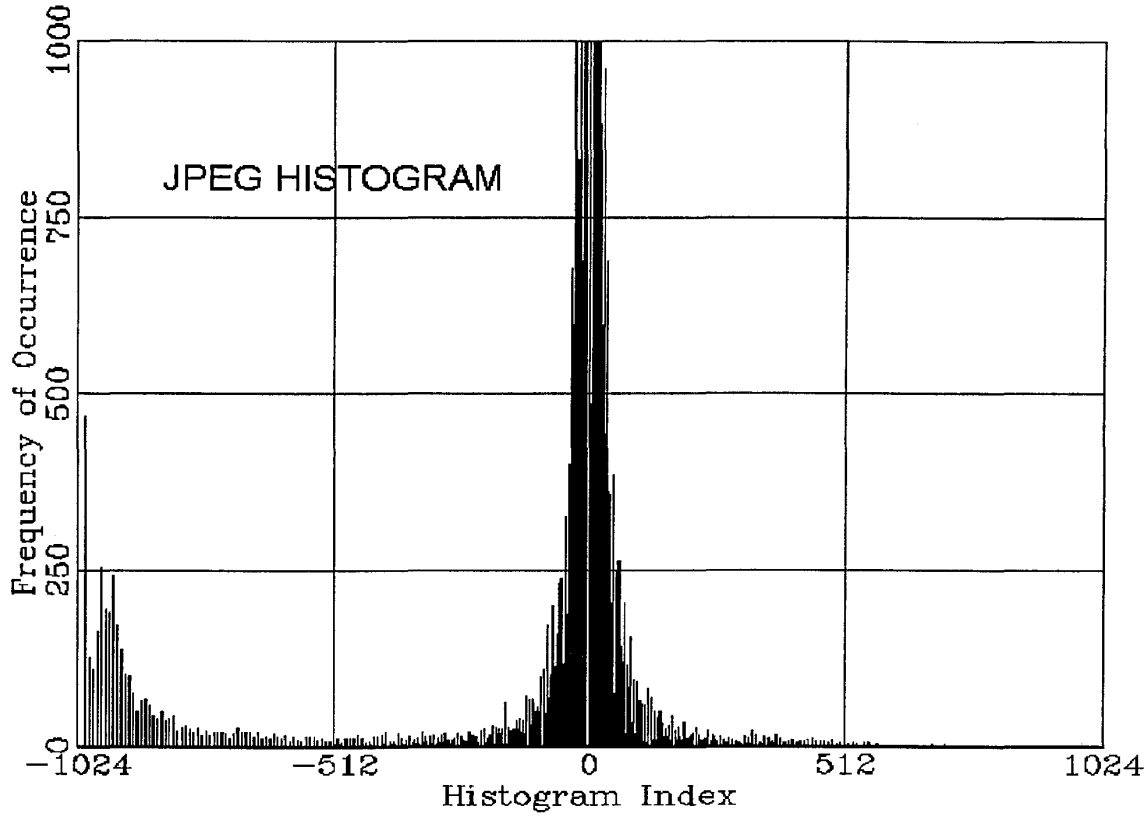

JPEG HISTOGRAM
Frequency of Occurrence
1000
750
500
250
0
−1024
−512
0
512
1024
Histogram Index

FIGURE 2

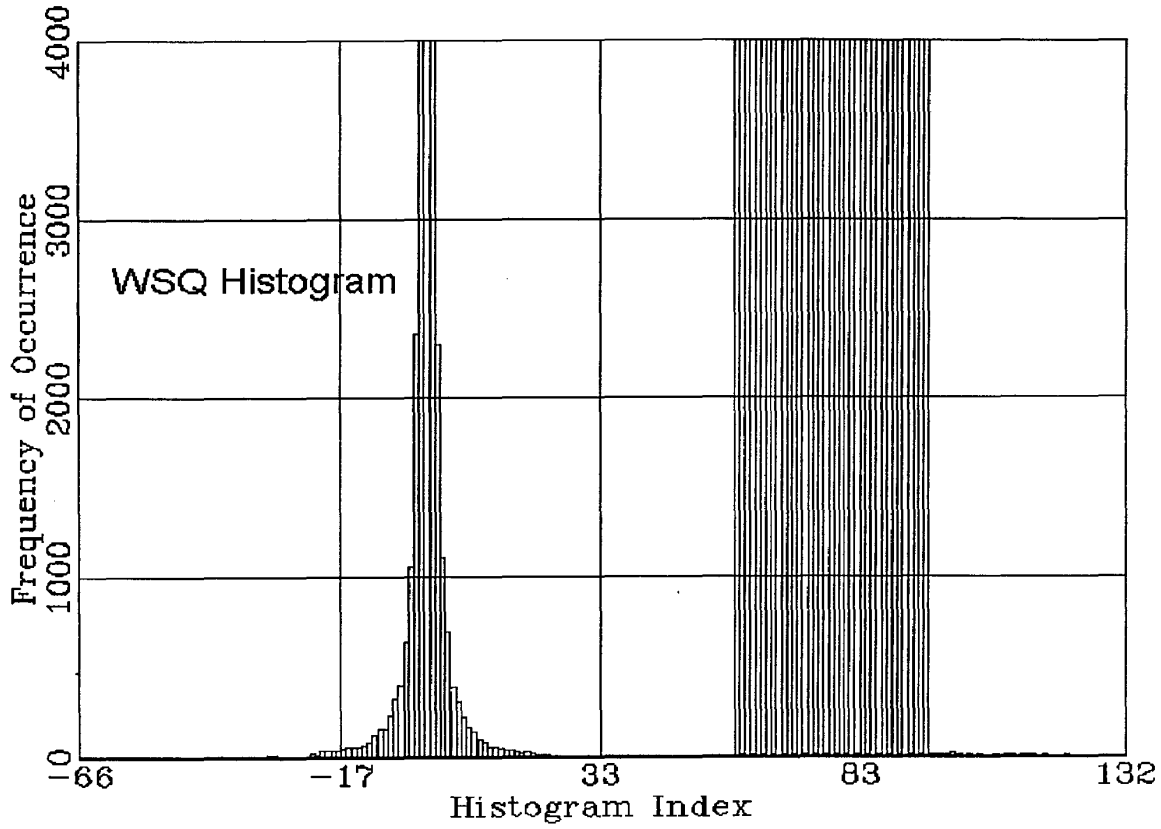

WSQ Histogram
Frequency of Occurrence
0
1000
2000
3000
4000
-66
-17
33
83
132
Histogram Index

FIGURE 3

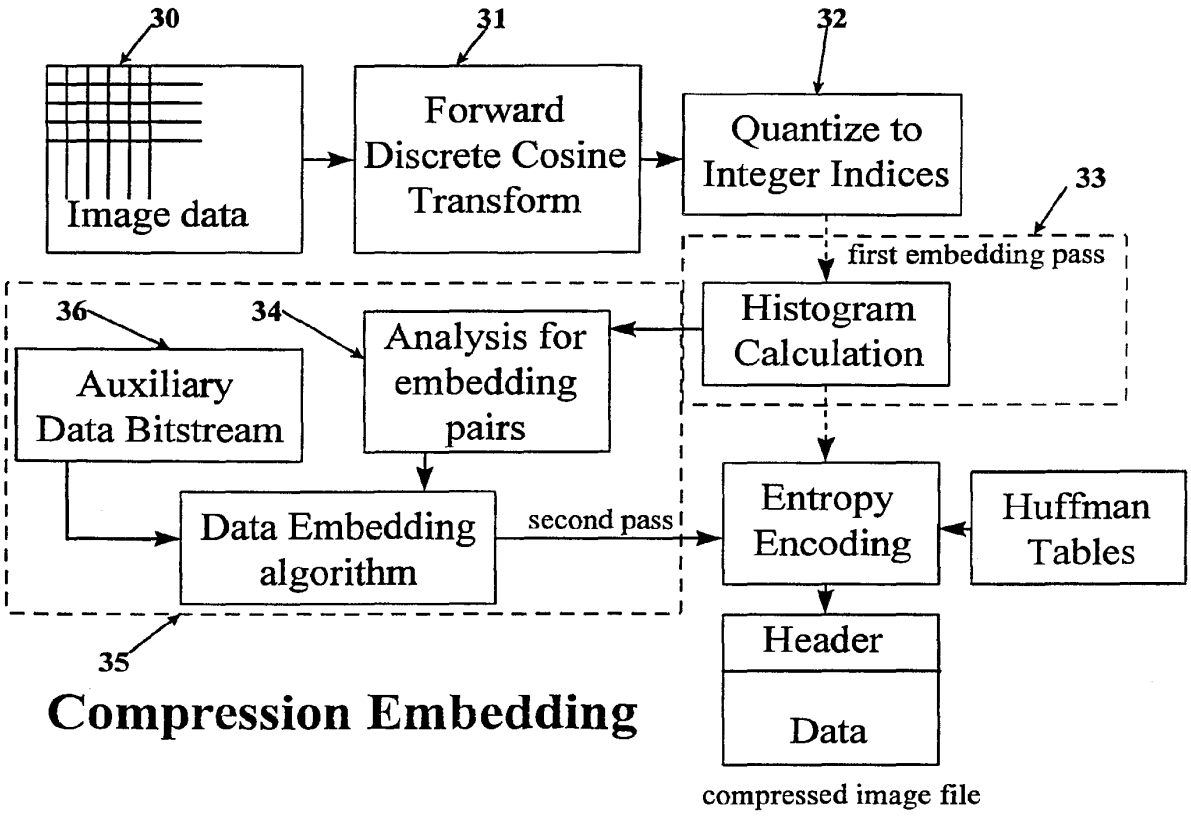

30
31
32
33
Image data
Forward Discrete Cosine Transform
Quantize to Integer Indices
first embedding pass
Histogram Calculation
36
34
Auxiliary Data Bitstream
Analysis for embedding pairs
Data Embedding algorithm
second pass
Entropy Encoding
Huffman Tables
Header
Data
compressed image file
35
Compression Embedding

FIGURE 4

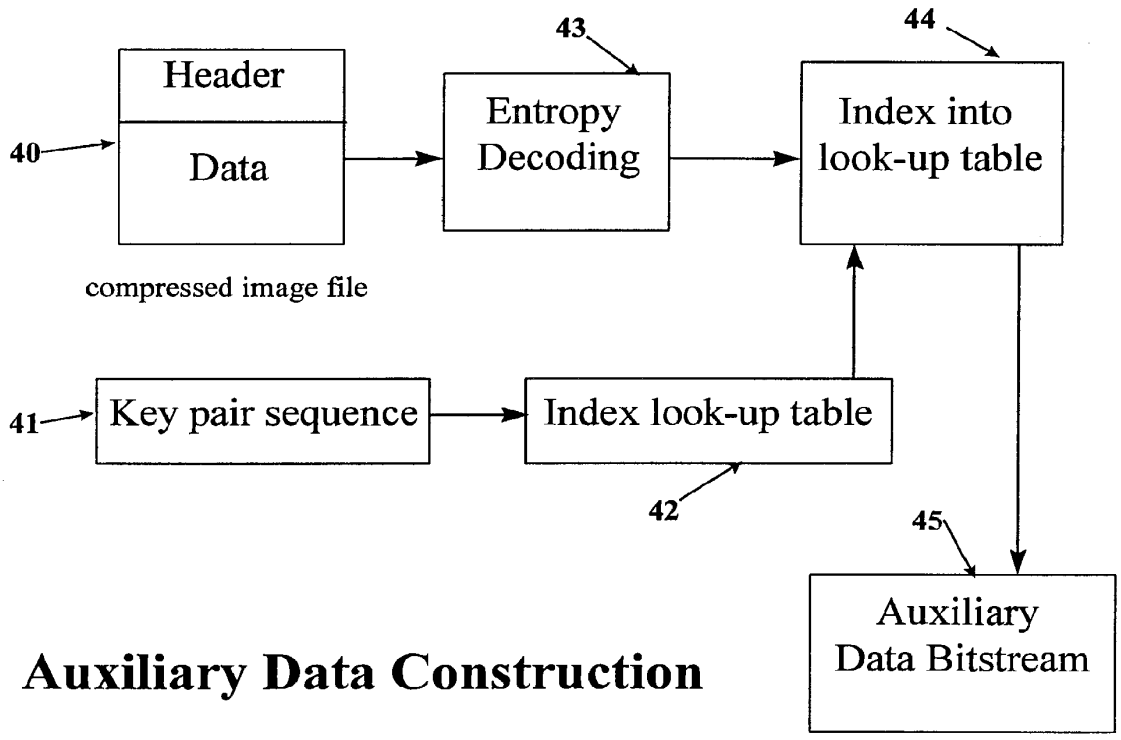

43
44
Header
Data
40
Entropy Decoding
Index into look-up table
compressed image file
41
Key pair sequence
Index look-up table
42
45
Auxiliary Data Bitstream
Auxiliary Data Construction

FIGURE 5

```
/*****************************************************************************************
* JHISTGM.C  Code module to process the histogram data to select key
*            pairs for use in
*            the compression embedding method.
*
* ORIGIN:  Derived from the Compression Embedding application code.
*
* Copyright (c) 1994-1996.  M. T. Sandford II and the University of California.
*     All Rights Reserved.
*
* Version 1.10  10-15-96
*****************************************************************************************/


#include <stdlib.h>
#include <stdio.h>
#include <string.h>

#define H_TABLE_SIZE 2048          /* range of the quantizer output (+/- 1023) */
#define MAXPAIRS 256      /* maximum number of pairs permitted for the key */
#define MINVAL -1024      /* minimum value from quantizer output */
#define HISTLIMIT  3                        /* low-frEquation limit for histogram
entries (+/- limit) */

struct PAIRS {
     short i;
     short j;
     unsigned long count;
     };


/* function prototypes */
void p_sort(struct PAIRS *pair, int no_pair);
void jumble(int *index, int npts);
int duplicate (int nopairs, struct PAIRS *pair);
void jhistgm(long *hist_table, struct PAIRS *pair,int *no_pairs);
void scramble(int no_pairs, struct PAIRS *pair);
extern unsigned long GetAuxFileSize(void);
extern unsigned short getrepwidth(unsigned long size);
```

FIGURE 6

```
/*****************************************************************************************
 * JHISTGM:  routine to process the histogram table and to determine a noise key
 *
 *    INPUT:                                    hist_table, pointer to unsigned long
histogram data
 *
 *    OUTPUT:   *pair[],    values of the pairs stored in a structure array
 *
 *    RETURN:   nothing

*****************************************************************************************
*/

void jhistgm(long *hist_table, struct PAIRS *pair,int *no_pairs)  {
int j;
int iloop;
long lsum;
long lavg;
long ldiff;
unsigned long file_size=0L;
/*    Process histogram to select embedding pairs */
  j = 0;
  for(iloop=0;iloop<H_TABLE_SIZE;iloop++)     {
      if(hist_table[iloop]==0 || abs(iloop+MINVAL) < HISTLIMIT)  {  //skip zero entries
& avoid low frEquation
                                                iloop += 1;
                                                continue;
                                                }
      lsum = hist_table[iloop]+hist_table[iloop+1];
      lavg = lsum/2L;
      ldiff = hist_table[iloop]-hist_table[iloop+1];
      ldiff = labs(ldiff);
      if(ldiff < lavg) {   /* Difference less than avg. for the pair */
                                                pair[j].i = iloop+(int)MINVAL;
                                                pair[j].j = iloop+(int)MINVAL+1;
                                                pair[j].count = (unsigned long)lsum;
                                                j++;
                                                iloop+=2;
                                                if(j==MAXPAIRS)break;
                                                }
      }
```

FIGURE 6

```
  lsum = 0L;
  *no_pairs = duplicate(j,pair);  /* remove duplicates (there should not be any) */
  for (iloop=0;iloop<*no_pairs;iloop++)  {
     if(pair[iloop].i==pair[iloop].j)break;
     lsum += pair[iloop].count;

#ifdef PROGRESS_REPORT
     fprintf(stderr,"\npair[%3d] %4d %4d %5ld", iloop,\
                                                pair[iloop].i, pair[iloop].j,
pair[iloop].count);
#endif
  }

  // check the embedding space against the amount of auxiliary data
  file_size = GetAuxFileSize();
  fprintf(stderr,"\n%d pairs located. Total embedding space = %ld bytes", *no_pairs,
lsum/8L);
  if(file_size*8L > (unsigned long)lsum)  {
     fprintf(stderr,"\nAuxiliary file size = %ld bytes", file_size);
     *no_pairs = 0;
     }

  // reduce key to the minimum size needed to embed the file
  for(iloop=0;iloop<*no_pairs;iloop++)  {
     if(lsum-pair[iloop].count >= file_size*8L + getrepwidth(file_size) + 5)  {
                                          lsum -= pair[iloop].count;
                                          pair[iloop].count = 0L;       // throw out this
pair
                                          }
     }
  *no_pairs = duplicate(*no_pairs,pair);   // remove discarded pairs

  scramble(*no_pairs,pair);                        // mix up the pairs to form a random
key
  lsum = 0L;
  for(iloop=0;iloop<*no_pairs;iloop++)
     lsum += pair[iloop].count;
  if((unsigned long)lsum >= file_size*8L + getrepwidth(file_size) +5)
     fprintf(stderr,"\n%d byte key",*no_pairs*2);
  return;
}
```

FIGURE 6

```
        /*    DUPLICATE:  Routine to remove entries containing duplicate
                          histogram values in the PAIRS structure
 *      INPUT:           no_pairs, integer no. of pairs in the pixel pair
                          table
 *                       pixpairs, table of PAIRS structures
 *      RETURN:          new value for no_pairs (0 if no entries exist)
 *      AUTHOR:     Copyright (c) 1994-95, M. T. Sandford II
 *                                                Univ. of California, Los Alamos
National
                          Laboratory
 ***************************************************************************/
#include <stdlib.h>

struct PAIRS {          /* structure of pallet pair data entries */
     int i;
     int j;
     unsigned long count;
     };
int duplicate (int nopairs, struct PAIRS *pair)  {
int i;        /* counters */
int j;
int k=0;
if(nopairs == 0)return k;                    /* return if nothing to do */
p_sort(pair,k);                              /* sort pairs by frequency of occurrence */
for(i=0;i<nopairs;i++)                       {
     if(pair[i].count==0L)continue;
     for(j=0;j<nopairs;j++)                  {
          if(j==i)continue;
          if(pair[j].count==0)continue;
          if(pair[j].i == pair[i].i ||
            pair[j].i == pair[i].j ||
            pair[j].j == pair[i].i ||
            pair[j].j == pair[i].j)  {
               if(pair[j].count < pair[i].count)
                    pair[j].count = 0;
               else {
                                        pair[i].count = 0;
                                        break;
                                        }
               }
          }
     }
```

FIGURE 7

```
for(i=0;i<nopairs;i++) { /* remove null entries from pair table */
      j = i+1;
      while(pair[i].count == 0) {
                                          for(k=j;k<nopairs;k++)
                                          pair[k-1] = pair[k];
                                          nopairs--;
                                          if(nopairs==0) break;
            }
     if(nopairs==0)break;
     }
return i;
}
```

FIGURE 7

```
/*****************************************************************************************
*
* SCRAMBLE.C:  Routine to scramble the pair order to a random sequence
*
* INPUT:      no_pairs, integer no. of pairs to scramble
*             pair, pointer to table of structures to scramble
*
* RETURNS:    nothing
*
* OUTPUT:     pair, pointer to the scrambled table of pair structures
*
* AUTHOR:     M. T. Sandford II, 6 Oct. 1994 * Version 1.0
* Copyright © 1996 The University of California and M. T. Sandford II
* All Rights Reserved.

*****************************************************************************************
/
void scramble(int no_pairs, struct PAIRS *pair) {
   int i;                                /* loop counter */
   int *index=NULL;                      /* scrambled index values */
   struct PAIRS *temp_pair=NULL;   /* temporary pair structure table */

   index = (int *)malloc(no_pairs*sizeof(int));
   temp_pair = (struct PAIRS *)malloc(no_pairs*sizeof(struct PAIRS));
   if(index==NULL || temp_pair==NULL) {
     if(index)free(index);
     if(temp_pair)free(temp_pair);
     fprintf(stderr,"\nJumble memory allocation failed!");
     return;   /* no scrambling performed */
     }
   jumble(index,no_pairs);
   for(i=0;i<no_pairs;i++) {
     temp_pair[i].i = pair[index[i]].i;
     temp_pair[i].j = pair[index[i]].j;
     temp_pair[i].count = pair[index[i]].count;
     }
   memmove(pair,temp_pair,no_pairs*sizeof(struct PAIRS));
   free(index);
   free(temp_pair);
```

FIGURE 8

```
   // now scramble the parity of the entries in the pair tables
   for(i=0;i<no_pairs;i++) {
     int temp;
     temp = rand();
     if(temp > RAND_MAX/2) {
                                          temp = pair[i].j;
                                          pair[i].j = pair[i].i;
                                          pair[i].i = temp;
                                          }
                                          }
   return;
}
/*****************************************************************************************
**
 * JUMBLE.C:  Routine to jumble a table of indices using pseudo-random
 *            numbers seeded from the PC clock
 *
 * INPUT:     index, pointer to a table of jumbled integers
 *          npts,  integer no. of entries in the index table
 *
 * RETURN:    nothing
 *
 * OUTPUT:    returns a jumbled table of integers
 *
 * AUTHOR:    M. T. Sandford II, 5 Oct. 1994, following the method
 *          preferred by T. Handel
 *****************************************************************************************/
void jumble(int *index, int npts) {
int i;
float fi;
struct POINTS {
   float gamma;
   int i;
   } *pts;
pts = malloc(sizeof(struct POINTS)*npts);
for (i=0;i<npts;i++) {
   pts[i].i = i;
   pts[i].gamma = (float)rand()/(float)RAND_MAX;
   }
qsort( (void *)pts, (size_t)npts, sizeof(struct POINTS),
                                          index_compare);
```

FIGURE 8

```
for(i=0;i<npts;i++) {
      index[i] = pts[i].i;
/*      printf("\ni,index[i] %d %d", i, index[i]); */
      }
if(index[0]==0) {
   fi = (float)rand()/(float)RAND_MAX;
   fi *= (float)(npts-1)+1.0f;
   index[0] = index[(int)fi];
   index[(int)fi] = 0;
   }
free(pts);
}
```

FIGURE 8

```
/*****************************************************************************
 * JEMBED.C  Code module to perform embedding into the quantizer indices
 *
 * Author:  M. T. Sandford II, Sept. 4, 1996.  Version 1.10
 *
 * Copyright (c) 1994-1996.  M. T. Sandford II and the University of California
 *   All Rights Reserved.
 *
*****************************************************************************/
#include <stdio.h>
#include <stdlib.h>
#include <memory.h>

#define MAXPAIRS 256                    /* maximum no. of pairs */
#define MAXSIZE  2049                   /* maximum no. of indices */
#define MINVAL  -1024      /* minimum index value */

/*****************************************************************************
 * Embed_Data_Block:  Routine to embed data into a block of integers using
 *                    the pairs in the structure pair.  This code was re-
 *                    written from the original version, to use table look-up
 *                    for the embedding values.
 *
 *   INPUT:             buffer,   pointer to the block of integers
 *   ibuf,              integer no. of integers in the block
 *
 *   Note:  The following are determined by reading the keyfile
 *                                  pair,    pointer to index pair structures
 *                                  no_pairs, number of pair structures
 *
 *   RETURN:                        nothing
 *
*****************************************************************************/

static int ptable[MAXSIZE];                 /* table of pairs */
static int ptable_filled = 0;               /* flag to fill the table */
static unsigned long embed_count=0L;  /* total number of embedded bits */
```

FIGURE 9

```
static struct PAIRS {                    /* structure of pallet pair data entries */
     int i;
     int j;
     unsigned long count;
     }pair[MAXPAIRS];
static int no_pairs;

extern short bitgetbit(void);        /* Derek Upham's data code */
extern FILE *fpKeyFile;                                          /* pointer to the
pair key file */

void Embed_Data_Block(short *buffer,int ibuf)  {
short index;
int i;
int j;
int k;
size_t inbytes=0;
if(!ptable_filled)  {
     fseek(fpKeyFile,0L,SEEK_SET);     // retrieve the pair values from the keyfile
     no_pairs = 0;
     for(i=0;i<MAXPAIRS;i++)  {
                                       k = sizeof(int)*fread(&pair[i].i,sizeof(int),1,
fpKeyFile);
                                       if(k==0)break;
                                       inbytes += k;
                                       k = sizeof(int)*fread(&pair[i].j,sizeof(int),1,
fpKeyFile);
                                       if(k==0)break;
                                       inbytes += k;
                                       no_pairs++;
                                       }
     pair[no_pairs].i = pair[no_pairs].j = 0;
     pair[no_pairs].count = 0L;
   fclose(fpKeyFile);                                    // all done with the
keyfile (it could be compressed now)
   fpKeyFile = NULL;
```

FIGURE 9

```
   for(k=0;k<MAXSIZE;k++) ptable[k] = -1;
     for(k=0;k<no_pairs;k++)  {
                                        index = pair[k].i - MINVAL;
                                        ptable[index] = k;
                                        index = pair[k].j - MINVAL;
                                        ptable[index] = k;
                                        }
     ptable_filled = 1;                                  // the key lookup table
is filled & can be used
   }


for(j=0;j<ibuf;j++)                     {           // loop over each value in the
buffer
                                        index = buffer[j]-MINVAL;
                                        k = ptable[index];
                                        if(k>=0) {          // does the value match
with a key-pair element?
                                                embed_count++;
                                                if ((i=bitgetbit()) != -1)  {
                                                        if(i) index = pair[k].j;
                                                        else  index = pair[k].i;
                                                        buffer[j] = index;
                                                        //buffer[j] = 35;           //
intentionally corrupt the element for display purposes
                                                        }
                                                }
                                        }
return;
}
```

FIGURE 9

```
/*****************************************************************************************
*   ExtractDataBlock:  Routine to extract data from a block of integers using
*                                          the pairs in the structure pair.  This code
was re-written from the
*                                          original version, to use table look-up for
the embedding values.
*
*   INPUT:                                 buffer,   pointer to the block of integers
*                                                ibuf,    integer no. of integers in the
block
*
*   Note:  The following are determined by reading the keyfile
*                                                pair,    pointer to index pair
structures
*                                                no_pairs, number of pair structures
*
*   RETURN:                                nothing
* Version 1.10 10/15/96
* Copyright © 1996 M. T. Sandford II and The University of California
* All Rights Reserved.
*****************************************************************************************/

static int ptable[MAXSIZE];                                    /* table of pairs */
static int ptable_filled = 0;                          /* flag to fill the table */
static unsigned long embed_count=0L;  /* total number of embedded bits */
static int no_pairs;
static int alldone=0;                                  /* flag for completion of
extraction */

extern struct PAIRS {                                  /* structure of pallet pair data entries
*/
    int i;
    int j;
    unsigned long count;
    }*pair;

extern int bitsetbit(unsigned char value);  /* Derek Upham's data code */
extern unsigned long GetFileSize(void);   /* routine to return extracted file size */
extern FILE *fpKeyFile;                        /* pointer to the pair key file */
```

FIGURE 10

```
void ExtractDataBlock(short *buffer,int ibuf) {
short index;
int i;
int j;
int k;
size_t inbytes=0;
unsigned long fsize = 0;

if(alldone) return;                          // nothing left to do

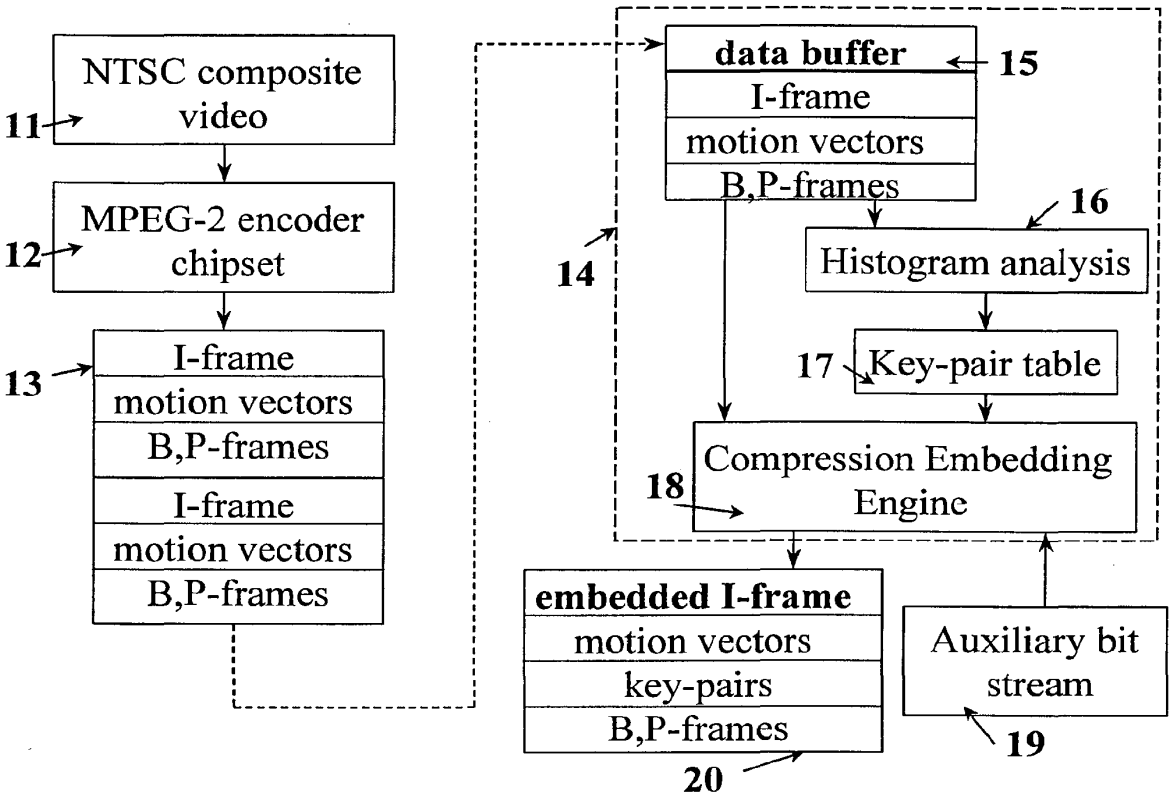
if(!ptable_filled) {
    //fpKeyFile = fopen("keyfile", "rb");    // retrieve the pair values from the keyfile
    no_pairs = 0;
    for(i=0;i<MAXPAIRS;i++) {
                                             k = sizeof(int)*fread(&pair[i].i,sizeof(int),1,
fpKeyFile);
                                             if(k==0)break;
                                             inbytes += k;
                                             k = sizeof(int)*fread(&pair[i].j,sizeof(int),1,
fpKeyFile);
                                             if(k==0)break;
                                             inbytes += k;
                                             no_pairs++;
                                             }
    pair[no_pairs].i = pair[no_pairs].j = 0;
    pair[no_pairs].count = 0L;
  fclose(fpKeyFile);                                    // all done with the keyfile
  fpKeyFile = NULL;
  for(k=0;k<MAXSIZE;k++) ptable[k] = -1;
    for(k=0;k<no_pairs;k++) {
                                             index = pair[k].i - MINVAL;
                                             ptable[index] = k;
                                             index = pair[k].j - MINVAL;
                                             ptable[index] = k;
                                             }
    ptable_filled = 1;                                  // the key lookup table is filled &
can be used
  }

for(j=0;j<ibuf;j++)                          {          // loop over each value in the
buffer
                                             index = buffer[j]-MINVAL;
                                             k = ptable[index];
                                             if(k>=0) {             // does the value match
with a key-pair element?
                                             if(pair[k].i == buffer[j]) i = bitsetbit(0x00);
                                             else i = bitsetbit(0x01);
```

FIGURE 10

```
                                        if(i==-1) {
                                                alldone = 1;
                                                fsize = GetFileSize();
                                                fprintf(stderr,"\n%6ld header
bytes extracted", embed_count/8L - fsize);
                                                fprintf(stderr,"\n%6ld   data
bytes extracted\n", fsize);
                                                fflush(stderr);
                                                }
                                        embed_count++;
                                        }
                                }
return;
}
```

FIGURE 10


NTSC composite video
11
MPEG-2 encoder chipset
12
I-frame
motion vectors
B,P-frames
I-frame
motion vectors
B,P-frames
13
14
data buffer
15
I-frame
motion vectors
B,P-frames
16
Histogram analysis
17
Key-pair table
Compression Embedding Engine
18
embedded I-frame
motion vectors
key-pairs
B,P-frames
20
Auxiliary bit stream
19

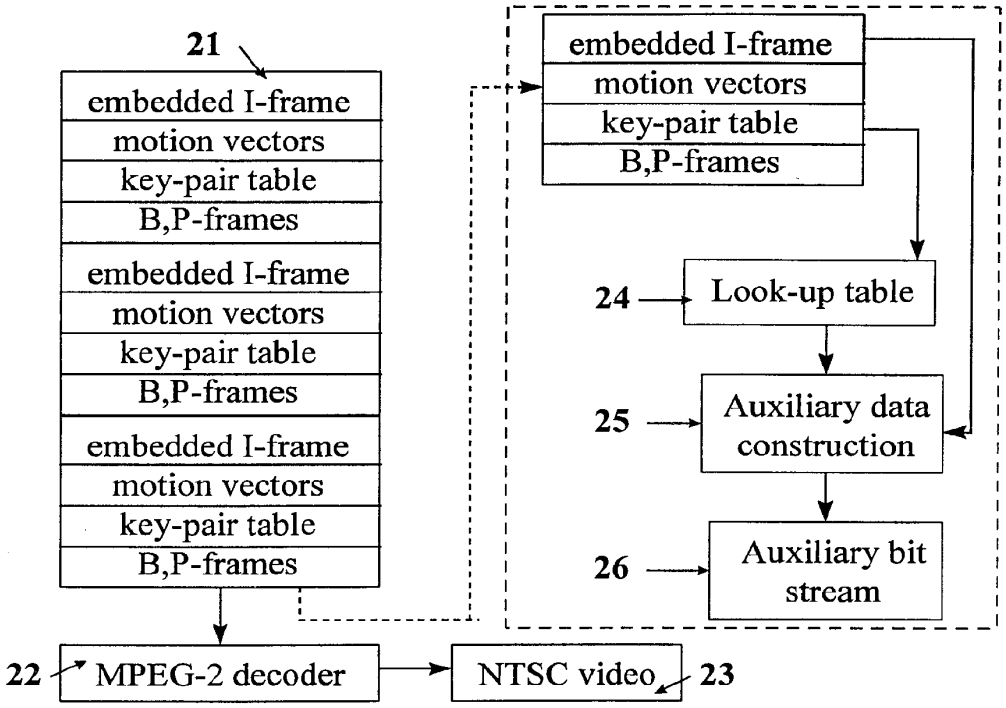

21
embedded I-frame
motion vectors
key-pair table
B,P-frames
embedded I-frame
motion vectors
key-pair table
B,P-frames
embedded I-frame
motion vectors
key-pair table
B,P-frames
22
MPEG-2 decoder
NTSC video
23
embedded I-frame
motion vectors
key-pair table
B,P-frames
24
Look-up table
25
Auxiliary data construction
26
Auxiliary bit stream

FIGURE 12

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US97/23291

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :G06K 9/00
US CL :Please See Extra Sheet.
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 364/715.02; 348/416; 235/382, 494; 358/426, 434; 382/232, 233, 236, 239, 243, 250, 254

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS
search terms: lossy, compression, integer

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A,P	US 5,636,292 A (RHOADS) 03 June 1997, column 3, lines 1-67.	1-29

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 23 FEBRUARY 1998	Date of mailing of the international search report 27 APR 1998
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer LEO BOUDREAU Telephone No. (703) 308-7595

Form PCT/ISA/210 (second sheet)(July 1992)★

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US97/23291

A. CLASSIFICATION OF SUBJECT MATTER:
US CL :

364/715.02; 348/416; 235/382, 494; 358/426, 434; 382/232, 233, 236, 239, 243, 250, 254

Form PCT/ISA/210 (extra sheet)(July 1992)★

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G11B 23/28, 23/36, 23/40	A1	(11) International Publication Number: **WO 00/36605**
		(43) International Publication Date: 22 June 2000 (22.06.00)

(21) International Application Number: PCT/SG98/00104

(22) International Filing Date: 11 December 1998 (11.12.98)

(71) Applicant *(for all designated States except US)*: KENT RIDGE DIGITAL LABS [SG/SG]; 21 Heng Mui Keng Terrace, Singapore 119613 (SG).

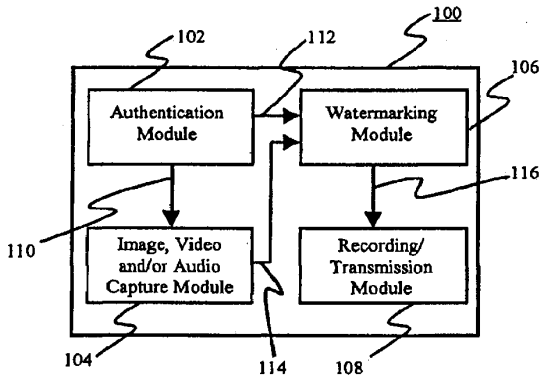
(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: WU, Jian, Kang [CN/SG]; Blk 51 Teban Gardens Road, #06–605, Singapore 600051 (SG). NARASIMHALU, A., Desai [IN/SG]; 103 Clementi Road, #03–01, Kent Vale, Singapore 129788 (SG). LO, Sheng [SG/SG]; Blk 444, Jurong West Avenue 1, #11–774, Singpaore 640444 (SG). LEBRUN, Jean–Luc [FR/SG]; 10G Braddell Hill, #23–27 Braddell View, Singapore 579726 (SG).

(74) Agent: SPRUSON & FERGUSON PTE LTD.; 51 Bras Basah Road, #02–03 Plaza by the Park, Singapore 189554 (SG).

(81) Designated States: GB, SG, US.

Published
With international search report.

(54) Title: METHOD AND DEVICE FOR GENERATING DIGITAL DATA WATERMARKED WITH AUTHENTICATION DATA



(57) Abstract

An apparatus (100), method and computer program product for generating an authenticated audio, image and/or video signal (116) is disclosed. The apparatus or recording device (100) includes an authentication module (102), an audio, image and/or video capture module (104), and a watermarking module (106). The apparatus can be a digital camera, digital video camera, or a digital audio recording device. The authentication module (102) checks if the operator is authorized to use the apparatus (100) and if so generates authentication information (112). Biometrics data such as a fingerprint, iris, voice and/or face recognition can be used for authentication. If authorized, the capture module (104) generates an audio, image and/or video signal (114). A watermarking module (106) embeds the authentication information (112) in the media signal (114) to provide the authenticated output signal (116). The watermarking is built into the recording device (100) and is carried out on–the–fly. The media signal can be watermarked with GPS data and/or annotation.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

METHOD AND DEVICE FOR GENERATING DIGITAL DATA WATERMARKED WITH AUTHENTICATION DATA

FIELD OF THE INVENTION

The present invention is directed to the field of recording digital audio, image and/or video signals and, in particular, to a method and apparatus for embedding information in digital audio, image and/or video signals.

BACKGROUND

The subsistence of copyright in artistic works is a significant and valuable right rewarding artistes for their original, creative efforts. Amongst other things, the subsistence of copyright in an artistic work allows a copyright owner to advantageously commercially exploit their work and also to control the use and treatment of such artistic works by others. The existence of copyright is particularly important in relation to artistic works in fields such as photography, music and film.

In conventional photography and other image acquisition processes, claims of ownership in photographs and similar image recordings are typically indicated by manually labeling the back of the photographs with details of the owner. Alternatively, stickers or rubber stamps may be used to mark the photographs. However, such methods do not provide incontrovertible or substantially incontrovertible evidence of ownership of the artistic work by the owner. For example, a false claim of ownership can be made on unmarked materials owned by another, can be defaced or erased, or can be overwritten with false markings of another claimant.

In film and video recordings, ownership rights are frequently made by adding details of the owner in a header and/or a trailer portion of the recording. For audio recordings, ownership details are normally indicated on the recording media (e.g., CD-ROM, tape, etc.) or packaging. However, ownership indicia are capable of being easily removed or altered from the recordings or packaging.

With the widespread advent of digital recording technology, artistic works can be readily duplicated and distributed worldwide immediately by communication channels like the Internet, Intranets, and satellite communications. Further, such digital artistic works are readily susceptible of unauthorised modification and tampering. The recording of an artistic work in the digital domain exacerbates the problems of establishing ownership of copyright in an artistic work. Several techniques have been attempted to address this issue.

United States Patent No. 5,513,260 discloses a method and apparatus that attempts to provide copyright protection for recording media such as compact discs (CDs) using encryption. In particular, a combination of symmetrical and asymmetrical data encryption is used to permit a reproduction device to handle either copy-protected or non-copy-protected media. The technique encrypts the digital recording on a recording media. However, the system does not directly relate the artistic work contained in the recording with the copyright owner, and a digital recording once decrypted can be copied or modified without any protection for the copyright owner.

United States Patent No. 4,890,319 discloses a subscription television system for distributing copyrighted program material to subscribers. The system has a transmitter with a data-insertion circuit for inserting a control bit in a transmitted signal, which comprises other control data and the copyrighted program material. A receiver in the system includes a memory for storing a pre-assigned subscriber identification. Upon receipt and identification of the control bit at the receiver, the receiver inserts a subscriber identification into the program material. An unauthorized copy of the program material may be determined since it includes the subscriber identification. However, again, the system does not directly relate the artistic work itself contained in the recording with the copyright owner, and a party familiar with the system can defeat the system by appropriate processing to remove the subscriber identification.

The foregoing systems are disadvantageous in that they fail to uniquely relate the content of the artistic work itself with the owner in a manner that effectively prevents

tampering. Further, the systems cannot ensure that the recordings to which the techniques are applied are the property of the party asserting ownership.

SUMMARY

The aspects of the invention are directed to ameliorating or overcoming one or more disadvantages of conventional systems including those described above. The aspects of the invention are able to do so by requiring authentication data derived from an authorised operator to operate a recording device and providing on-the-fly copyright ownership indicia in a recording, when recording, by watermarking the recording using the authentication data.

In accordance with a first aspect of the invention, a method of generating an authenticated audio, image and/or video signal is disclosed. The method includes the steps of generating an authentication signal, capturing a digital audio, image and/or video signal, and watermarking the digital audio, image audio, image and/or video signal using the authentication signal to provide the authenticated audio, image and/or video signal.

Preferably, the method further includes the steps of: determining if an operator is authorised to generate the authenticated audio, image and/or video signal; and if so, enabling capture of the digital audio, image and/or video signal.

Preferably, the step of generating the authentication signal is performed substantially simultaneously with the step of capturing the digital audio, image and/or video signal. Further, the authentication signal can be derived from biometrics data, a password, user identification data, positioning data, chronological data, data transmitted from an external, remote source and/or data from a removable media.

Optionally, the step of capturing the digital audio, image and/or video signal includes: capturing an analog audio, image and/or video signal; and converting the analog audio, image and/or video signal to the digital audio, image and/or signal.

Also, the watermarking step provides perceptible and/or non-perceptible watermarks. Optionally, the method includes annotating the digital audio, image and/or video signal using text and/or audio data. The annotation can be persistent and/or non-persistent.

In accordance with a second aspect of the invention, an apparatus for generating an authenticated audio, image and/or video signal is disclosed. The apparatus includes: a module for generating an authentication signal; a module for capturing a digital audio, image and/or video signal; and a module for watermarking the digital audio, image and/or video signal using the authentication signal to provide the authenticated, copyrighted audio, image and/or video signal.

Preferably, the apparatus further includes: a module for determining if an operator is authorised to generate the authenticated audio, image and/or video signal; and a module for enabling capture of said digital audio, image and/or video signal in response to determining the operator is authorised.

Preferably, the module for capturing the digital audio, image and/or video signal operates substantially simultaneously with the module for generating the authentication signal. Further, the authentication signal can be derived from biometrics data, a password, user identification data, positioning data, chronological data, data transmitted from an external, remote source and/or data from a removable media.

Optionally, the module for capturing the digital audio, image and/or video signal includes: a module for capturing an analog audio, image and/or video signal; and a module for converting the analog audio, image and/or video signal to the digital audio, image and/or signal.

Also, the watermarking module provides perceptible and/or non-perceptible watermarks. Optionally, the apparatus includes a module for annotating the digital

audio, image and/or video signal using text and/or audio data. The annotation can be persistent and/or non-persistent.

In accordance with a third aspect of the invention, a computer program product having a computer readable medium having a computer program recorded therein for generating an authenticated audio, image and/or video signal is disclosed. The computer program product includes: means for generating an authentication signal; means for capturing a digital audio, image and/or video signal; and means for watermarking the digital audio, image and/or video signal using the authentication signal to provide the authenticated audio, image and/or video signal.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention are described hereinafter with reference to the drawings, in which:

Figure 1 is a block diagram of an apparatus for generating an authenticated audio, image and/or video signal in accordance with a first embodiment of the invention;

Figure 2 is a flowchart illustrating a method of generating an authenticated audio, image and/or video signal in accordance with a second embodiment of the invention; and

Figure 3 is a block diagram of an apparatus for generating an authenticated audio, image and/or video signal in accordance with a third embodiment of the invention;

DETAILED DESCRIPTION

A method, apparatus and computer program product for generating an authenticated audio, image and/or video signal, where the authenticated signal is an audio, image and/or video signal watermarked with authentication data, are described. In the following description of several embodiments, numerous specific details such as particular watermarking techniques are described in order to provide a more thorough

description of those embodiments. It will be apparent, however, to one skilled in the art that the present invention may be practiced without those specific details. In other instances, well-known features such as particular video formats (e.g. MPEG), particular biometrics data, for example, have not been described in detail so as not to obscure the invention.

In broad terms, the embodiments of the invention provide a mechanism for providing a robust, tamper-evident "stamp" of authorship and/or ownership of audio, image and video works. This is done by requiring authentication data to operate an audio, image and/or video-recording device and providing on-the-fly copyright ownership indicia in an audio, image and/or video recording, when recording, by watermarking the recording using the authentication data. The authenticated watermarking is used to verifiably establish the subsistence of copyright ownership of the audio, image and/or video materials.

In the embodiments of the invention, both the generation of authentication data specific to the owner or operator of the recording device and the watermarking are implemented as components of the recording device. Amongst other things, this overcomes disadvantages of the prior art where watermarking might be implemented as an external post-processing procedure on a work downloaded or retrieved from a recording device at one point in time and then watermarked at a later point. In such circumstances, the presence of an original un-watermarked audio, image and/or video work external to the recording device provides opportunities for further disputes in the authenticity of the images. This might occur for example where a third party obtained a copy of the original un-watermarked audio, image and/or video work and watermarked it with data indicating that it was their property. In such circumstances, both the proper owner and the third party would have watermarked copies of the original work with conflicting claims of authorship and/or ownership.

The embodiments of the invention embed watermarking technology directly into the audio, image and/or video acquisition process as a means of protecting intellectual

property rights in the acquired work. In particular, one embodiment directly embeds the watermarking module into the hardware of the recording or acquisition device. Such an arrangement can render every segment of captured media via the recording device to bear the desired digital signature and/or caption embedded by the watermarking module.

For example, when a digital photo or a video segment is taken, immediately there can be potential copyright and annotation issues. Examples of persons affected by such issues include professional journalists and film producers. The embodiments of the invention significantly reduce and potentially eliminate conflicts regarding copyright authorship and ownership if the photo and video, for example, are watermarked with authenticated information specific to the journalist and film producer as soon as they are created. With a properly watermarked photo, the journalist can safely send it to the person's news agent and other news agents with significantly less concern that such persons might successfully assert an improper claim of copyright and/or ownership of the work. Since the audio, image and/or video work is in digital form, it can be easily indexed, stored in a data storage device, and transmitted over a network.

To prevent unauthorized access of the camera/recorder and unauthorized use of data captured by the camera/recorder, an authentication module is included in the camera/recorder to control operation of the camera/recorder. The authentication module may implement any of a number of authentication techniques such as: biometrics data including fingerprints, voice, and face of an authorised operator; a removable smart card containing encoded or encrypted data specific to the authorised operator; a password known by the authorised operator, etc. Alternatively, access data may be transmitted from a remote device such as radio, infrared or other appropriate transmitter acting as a security key of sorts. The authentication module further serves to enforce the authenticity of the recording equipment. The embodiments of the invention can be implemented to render the recording equipment useless in the event of theft, for example.

Optionally, the embodiments of the invention may also incorporate encryption and scrambling techniques, which are widely used in relation to digital photographs, video, and audio to protect them from unauthorized usage. Still further, the embodiments may incorporate an annotation module for annotating an audio, image and/or video work with text and/or audio information, for example. Optionally, the embodiments of the invention implement persistent annotation using watermarking and on-line annotation using voice.

A block diagram of a recording device according to a first embodiment of the invention is depicted in Fig.1. The device 100 for generating an authenticated audio, image and/or video signal includes an authentication module 102, an image, video and/or audio capture module 104, and a watermarking module 106 as its principal features. It also includes a recording/ transmission module 108 for recording and/or transmitting the authenticated audio, image and/or video signal containing embedded watermarks. The authentication module 102 is used to determine whether or not an attempted operator of the device 100 is an authorized user. It is also used to obtain and determine authentication information 112, which is subsequently watermarked in the capture audio, image and/or video work. The capture module 104 can be a type of digital camera, video camera or audio recorder. The watermarking module 106 embeds the authentication information 112 into the digital media 114 to provide the authenticated audio, image and/or video signal 116. Again, this is done to indicate copyright authorship and/or ownership of the audio, image and/or video work.

In particular, the authentication module 102 provides an enabling/disabling signal 110 to the image, video and/or audio capture module 104 dependent upon the results of checking if the attempted operator is authorised to use the device 100. For example, if the authentication module 102 determines that the attempted operator is authorised to operate the device 100, it generates an enable signal 110 which can actuate the image, video and/or audio capture module 104 to generate a relevant signal. Conversely, the authentication module 102 can disable 110 the capture module 104 if the attempted operator is not authorised.

The authentication module 102 protects usage of the device 100 as well as creation of authentication information 112. Authentication can be implemented in any of a number of ways. The simplest is to require the operator to input a password, perhaps using a keypad, to the recording device 100. Another way of implementing authentication is to require the user to swipe an encoded magnetic card through a card reader built in as part of the recording device 100. Alternatively, the device 100 can be adapted to receive an electronically readable card such as a smart card serving as a token that the authentication module 102 can check. Perhaps, the most secure way is to use a device for capturing biometric data from the operator such as fingerprint, facial, or speech recognition. As an example, for the case where the recording device 100 is a digital camera, a transparent button may be used as the button for actuating the digital camera to capture an image in a conventional manner. The authentication module may include a sensor for capturing the fingerprint of the operator when the operator presses the camera actuation button. Alternatively, the sensor may be oppositely disposed in the body of the recording deive, such as at its bottom, where the operator might place a thumb when pressing the actuation button.

Prior to any other processing by the camera, the authentication module 102 can check the captured fingerprint against one or more authorised fingerprints stored in a storage device (e.g. an electronic memory, magnetic storage device, optical storage device,) forming part of the authentication module 102 to determine if the operator is authorised to use the camera. Another implentation may obtain a retinal scan of an eye of the operator as biometric data when the operator looks into the camera sight and presses the actuation button. Still another implementation may use speech recognition in respect of an audio recorder to actuate the recorder where the recorder is voice actuated. As specific details for obtaining biometric data and comparing them with stored data are well known, these details of the authentication module 102 are not set forth in further detail so as not to obscure the invention. Likewise the use of passwords, magnetic cards and electronic cards such as smart cards are well known and therefore these features are not set forth in greater detail.

Authentication carried out by the authentication module 102 can be performed when the operator starts using the recording device 100, and when the operator stops using it for a certain period. Alternatively, the authentication module 102 may require verification of the operator periodically during operation of the recording device 100. For example, for each shut of a digital camera, the authentication module 102, if implemented as an iris-recognition module or a fingerprint-recognition module, can be built in to verify automatically the operator when that person aims the digital camera at a target.

Preferably, default authentication information 112 is loaded each time an authorised operator starts using the recording device 100. Optionally, an editing function is provided to enable the operator to edit authentication information contained in or derived from the authentication module 102. The default authentication information 112 may include one or more of the following: the name of the owner of the recording device 100 or of the authorised operator, the organization relevant to the owner or operator, and the time and date. The foregoing authetication information is merely exemplary. Alone or in combination with other data, it may contain original biometric data such as information derived from the operator's fingerprint. As described hereinafter in relation to another embodiment of the invention hereinafter, the authentication information 112 may optionally include location data identifying the place where the media data is captured. The location may be automatically determined using the Global Position System (GPS).

As yet another example, a recording device 100 according to the first embodiment can be implemented as a high-end digital camera, which is owned by a journalist "John" who is the only authorised user of the camera. In the camera as purchased, a password can be set in the authentication module 102. John can use the password to access the camera 100 initially, change the password to be his fingerprint, and register himself as the only authorised user. He can also set the default authentication information 112 to include his full name, time and location when the picture is taken. The authentication

module 102 can include a small-size, solid-state fingerprint scanner (not shown) that can be well fitted readily to the body of the digital camera 100. When John holds the camera 100 ready to capture an image, his thumb is on the scanner. Without John noticing, when John is holding the camera and attempts to actuate the shutter button, the scanner captures his thumb print and the thumb print is processed by the authentication module 102 to extract minuature points from the scanned image. For detailed information about biometrics authentication, reference is made to Jain, A., Bolle, R., and Pankanti, S., BIOMETRICS: Personal Identification in Networked Society, Kluwer Academic Publishers, October 1998. The minuature points can then be matched by the authentication module 102 against ones captured and stored together with his name and other authentication information in a storage device of the capture module 102. The match can take account of the location, type, and orientation of those minuature points. The default authentication information 112 can be retrieved and sent to the watermarking module 106. If the authentication is not successful, operation of the camera shutter in the image capture module 104 is locked or inhibited.

For each individual operation of the recording device 100 or at predetermined instances in a continuous recording interval, the captured media data 114 output by the capture module 104 is watermarked automatically by the watermarking module 106 by embedding one or more watermarks into the media data 114. Preferably, each watermark is a non-perceivable signal. However, the watermarking module 106 can be implemented to embed perceivable watermarks. The embedded signal preferably contains the authentication information the user has edited. Any of a number of watermark techniques for image, audio and video signals may be practiced in the watermarking module 106. Preferably, the particular watermarking techniques to be practiced in the embodiments of the invention possess the following features: non-perceivable, robust, and tamper-proof. By "non-perceivable" is meant that there is no perceivable difference between an original and a watermarked media (i.e., audio, image and/or video signal). In other words, the watermarking process does not affect a predetermined level of humanly perceivable quality of the original media. By "robustness" is meant that an embedded watermark signal is not removable under

media processing and manipulation processes such as filtering, analog-to-digital (A/D) and digital-to-analog (D/A) conversion, and geometric transformations. By "tamper-proof" is meant that the ownership of the media is well-protected against attacks by others.

For additional security, the watermarked media signal can be scrambled/encrypted using a private key generated from the authentication information, for example, or directly from the biometrics data.

As a further optional feature, a database entry can be created in a storage medium of the recording device 100 to keep a record of watermarked media data. For law-enforcement purposes, amongst others, certain registration information contained in the database can be sent to an authority or a public service center, in a manner known to those skilled in the art. Similar techniques are used to register digital signatures via the Internet, for example. This registration information can later be used to verify the authorship and/or ownership of the audio, image and/or video work that has been watermarked with authentication data.

Again with reference to an implementation of the recording device 100 as a digital camera (the following applies equally to digital video and audio recording), after authentication that may be transparent to the operator, John identifies new scenery for image capture and presses the shutter button of the camera 100. Actuation of the button activates capture of a digital photo by the capture module 104 and watermarks the digital photo or image 114 with the authentication information 112 provided by authentication module 102. There are many image watermarking techniques available for such a digital photo application. For an exemplary watermarking technique, reference is made to Cox, I.J., Kilian, J., Leighton, T. and Shamoon, T., Secure Spread Spectrum Watermarking for Multimedia, NEC Technical Report, 95-10. That watermarking technique can be used in this embodiment to convert the authentication information (e.g., the operator's identification) into signal patterns based on the concept of spread spectrum communication. The signal patterns are added to the

original image in the transform domain (e.g., digital cosine transform can be used). By doing so, the embedded watermark signal is spread over the whole image of the digital photo 114. It is therefore difficult to remove the watermark signal using ordinary image processing operations, making it strongly tamper-proof.

Further, the digital photo watermarked with the authentication data 116 output by the watermarking module 106 can be recorded on a storage medium or transmitted to an external location by the recording/transmission module 108.

The recording/transmission module 108 can include storage media such as electronic, magnetic, optical and/or magneto-optical storage media, which can be implemented as removable or non-removable storage media. As an illustrative example, the media can be a removable storage medium such as a floppy disc, which is used in a number of commercial digital cameras. Numerous other media can be practiced without departing from the scope and spirit of the invention. The recording/transmission module 108 can be implemented to transmit the output signal 116 to an external device or location. For example, the output signal 116 can be transmitted by an infra-red transmitter built into the recording device 100 to a corresponding receiver in another device such as a computer. Alternatively, it can be transmitted electrically via an appropriate interface built into the device 100 via one or more conductors to an external device and then to the Internet or an Intranet. The foregoing are merely indicate a few of the many ways of transmitting information between two devices that can be practiced. The watermarked photo 116 can be sent to John's representative so that the photo can be immediately used by various news agents. Preferably, the communication of the photo from the recording/transmission module 108 is via a wireless communication network. Optionally, registration information about the digital photo 116 can also be sent to a service center if necessary.

In the foregoing embodiment of the invention, the functionality of each module 102, 104, 106, and 108 can be implemented as an electronic circuit. Alternatively, at least part of the functionality of each module may be implemented as a computer program

that is carried out by a microprocessor coupled to appropriate storage means, both forming part of the recording device 100. Numerous audio, image and video capture devices are implemented utilising microprocessors and other forms of computer systems. For example, the watermarking module 106 can be implemented entirely as software executing on a microprocessor in the recording device 100. Analogously, the authentication module can be implemented using an appropriate sensing or coupling device to obtain a required external signal or information in conjunction with software executing on a microprocessor to carry out the remaining functionality of the module or a portion thereof.

A flowchart illustrating the method for generating an authenticated audio, image and/or video signal in accordance with the second embodiment is depicted in Fig. 2. Processing commences in step 200. In step 202, a check is made to determine if the operator is authorised, and if so authentication data is generated. In step 204, image, video, and/or audio data is captured if the operator is authorised. Steps 202 and 204 may be performed substantially simultaneously without departing from the scope and spirit of the invention. In step 206, the captured audio, image and/or video signal is watermarked with the authentication data. For further details of specific steps to be carried out, reference is made to the functionality of the apparatus of the first embodiment described above with reference to Fig. 1.

An apparatus in accordance with a third embodiment of the invention is depicted in Fig. 3. Features of the first embodiment that have corresponding parts in the third embodiment are denoted with a corresponding reference numeral. Thus, the authentication module 102 of Fig. 1 is denoted with reference numeral 302 in Fig. 3. For the purpose of brevity in the description, reference is simply made to the description of those corresponding modules in the description hereinbefore and not repeated here. Only the new elements of Fig. 3 are described in detail hereinafter. While the third embodiment illustrates two additional modules 318 and 320, it will be appreciated by one skilled in the art that the embodiment can be practiced optionally with only one of the two modules.

The recording device 300 includes a global positioning system (GPS) module 320 that can obtain accurate positioning information about the location of the recording device 324 when in operation. The GPS data 324 may be obtained at a predetermined instance or interval to be embedded in the captured signal 314 by the watermarking module 306. For example, after the operator is authenticated as authorised to operate the recording device 300 and when the capture module 304 is actuated, the GPS module 320 may obtain fresh GPS data or retrieve most recently stored GPS data to be provided as output 324 to the watermarking module 306.

Further, the recording device 300 can include an annotation module 318. The annotation module annotation module 318 can add annotation text, voice, or links to other media-segments directly into the media signal produced by the capture module 304 in a conventional manner. The annotation can be persistent and/or non-persistent. For example, a microphone and/or text input device can be included in the recording device 300 as part of the annotation module 318 for annotation purpose. Annotation information can be stored together with the media data based on a standard data format. For example, in the case of video media, the MPEG2 standard can be used which provides space for annotation. This is non-persistent annotation. Alternatively or in addition to non-persistent annotation, persistent annotation can be incorporated into the captured media signal 314.

The annotation module 318 also provides annotation signal 322 to the watermarking module 306. To make the annotation persistent, watermarking is used again to embed the annotation information 322 into the media data 314. Such persistent annotation can be either perceivable or non-perceivable. For example, persistent annotation is useful to put particular information at a particular location of an image. Alternatively, it can be used to put information in a particular interval of audio for indication of information like "this is John" or "sang by Linda". Therefore, the persistent and non-persistent annotation is useful to specify messages to be embedded in a captured media

signal, as well as information such as the location or time period which can be embedded in the media signal.

The foregoing embodiments of the invention disclose a method, apparatus and computer program product for generating an authenticated audio, image and/or video signal, where the authenticated signal is an audio, image and/or video signal watermarked with authentication data. The embodiments of the invention are advantageous in that they ameliorate or overcome one or more disadvantages of conventional audio, image and video capture systems. In particular, the embodiments of the invention are able to do so by requiring authentication data derived from an operator to operate a recording device and providing on-the-fly copyright ownership indicia in a recording by watermarking the recording using the authentication data. This is done in the recording device that is used to capture the audio, image and/or video work.

A small number of embodiments have been disclosed by way of example. However, those skilled in the art will recognise that the invention can be practiced, with modification, in the light of the information contained herein without departing from the scope and spirit of the invention.

The claims defining the invention are as follows:

1. A method of generating an authenticated audio, image and/or video signal, including:

generating an authentication signal;

capturing a digital audio, image and/or video signal; and

watermarking said digital audio, image and/or video signal using said authentication signal to provide said authenticated audio, image and/or video signal.

2. The method according to claim 1, wherein said step of generating said authentication signal is performed substantially simultaneously with said step of capturing said digital audio, image and/or video signal.

3. The method according to claim 1 or 2, wherein said authentication signal is derived from biometrics data, a password, user identification data, positioning data, chronological data, data transmitted from an external, remote source and/or data from a removable media.

4. The method according to any one of claims 1 to 3, wherein said step of capturing said digital audio, image and/or video signal includes:

capturing an analog audio, image and/or video signal; and

converting said analog audio, image and/or video signal to said digital audio, image and/or signal.

5. The method according to any one of claims 1 to 4, wherein said watermarking step provides perceptible and/or non-perceptible watermarks.

6. The method according to any one of claims 1 to 5, further including the step of annotating said digital audio, image and/or video signal using text and/or audio data.

7. The method according to any one of claims 1 to 6, wherein said annotating step is persistent and/or non-persistent.

8. The method according to any one of claims 1 to 7, further including the steps of:

determining if an operator is authorised to generate said authenticated audio, image and/or video signal; and

if so, enabling capture of said digital audio, image and/or video signal.

9. An apparatus for generating an authenticated audio, image and/or video signal, including:

means for generating an authentication signal;

means for capturing a digital audio, image and/or video signal; and

means for watermarking said digital audio, image and/or video signal using said authentication signal to provide said authenticated audio, image and/or video signal.

10. The apparatus according to claim 9, wherein said means for generating said authentication signal operates substantially simultaneously with said means for capturing said digital audio, image and/or video signal.

11. The apparatus according to claim 9 or 10, wherein said authentication signal is derived from biometrics data, a password, user identification data, positioning data, chronological data, data transmitted from an external, remote source and/or data from a removable media.

12. The apparatus according to any one of claims 9 to 11, wherein said means for capturing said digital audio, image and/or video signal includes:

means for capturing an analog audio, image and/or video signal; and

means for converting said analog audio, image and/or video signal to said digital audio, image and/or signal.

13. The apparatus according to any one of claims 9 to 12, wherein said means for watermarking provides perceptible and/or non-perceptible watermarks.

14. The apparatus according to any one of claims 9 to 13, further including means for annotating said digital audio, image and/or video signal using text and/or audio data.

15. The apparatus according to any one of claims 9 to 14, wherein said means for annotating provides persistent and/or non-persistent annotation.

16. The apparatus according to any one of claims 9 to 15, further including:

means for determining if an operator is authorised to generate said authenticated audio, image and/or video signal; and

means for enabling capture of said digital audio, image and/or video signal in response to determining said operator is authorised.

17. A computer program product having a computer readable medium having a computer program recorded therein for generating an authenticated audio, image and/or video signal, said computer program product including:

means for generating an authentication signal;

means for capturing a digital audio, image and/or video signal; and

means for watermarking said digital audio, image and/or video signal using said authentication signal to provide said authenticated audio, image and/or video signal.

18. The computer program product according to claim 17, wherein said means for generating said authentication signal operates substantially simultaneously with said means for capturing said digital audio, image and/or video signal.

19. The computer program product according to claim 17 or 18, wherein said authentication signal is derived from biometrics data, a password, user identification

data, positioning data, chronological data, data transmitted from an external, remote source and/or data from a removable media.

20. The computer program product according to any one of claims 17 to 19, wherein said means for capturing said digital audio, image and/or video signal includes:

means for capturing an analog audio, image and/or video signal; and

means for converting said analog audio, image and/or video signal to said digital audio, image and/or signal.

21. The computer program product according to any one of claims 17 to 20, wherein said means for watermarking provides perceptible and/or non-perceptible watermarks.

22. The computer program product according to any one of claims 17 to 21, further including means for annotating said digital audio, image and/or video signal using text and/or audio data.

23. The computer program product according to any one of claims 17 to 22, wherein said means for annotating provides persistent and/or non-persistent annotation.

24. The computer program product according to any one of claims 17 to 23, further including:

means for determining if an operator is authorised to generate said authenticated audio, image and/or video signal; and

means for enabling capture of said digital audio, image and/or video signal in response to determining said operator is authorised.

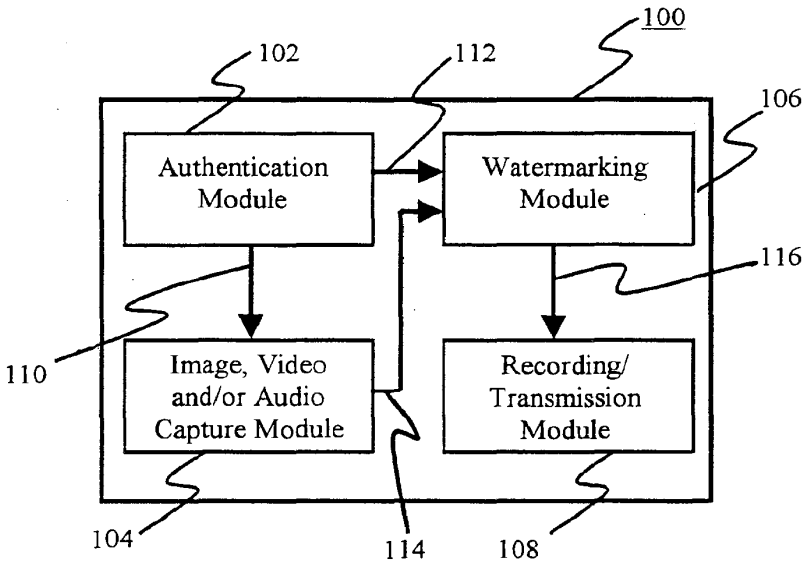

100
102
112
106
Authentication Module
Watermarking Module
116
110
Image, Video and/or Audio Capture Module
Recording/ Transmission Module
104
114
108

FIG. 1

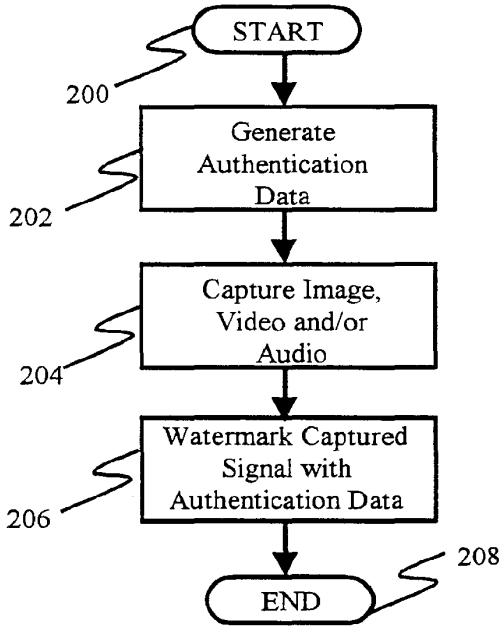

START
200
Generate Authentication Data
202
Capture Image, Video and/or Audio
204
Watermark Captured Signal with Authentication Data
206
208
END

FIG. 2

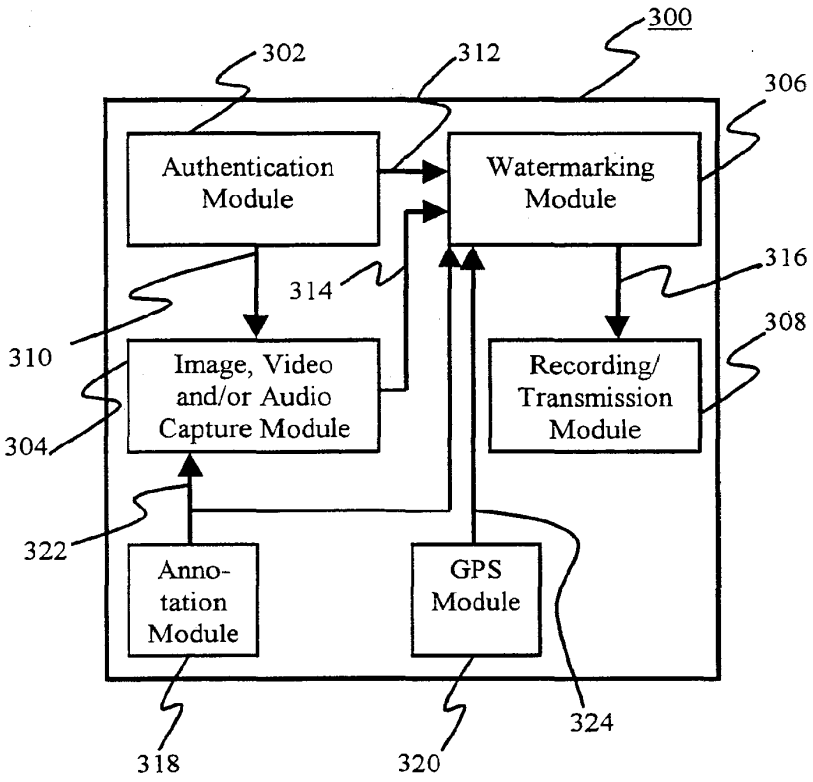

300
302
312
306
Authentication Module
Watermarking Module
316
314
310
308
Image, Video and/or Audio Capture Module
Recording/ Transmission Module
304
322
Anno-tation Module
GPS Module
324
318
320

FIG. 3

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SG 98/00104

A. CLASSIFICATION OF SUBJECT MATTER

IPC[6]: G 11 B 23/28, 23/36, 23/40

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC[6]: G 11 B 23/00; H 04 N 5/00, 7/00; G 07 C 9/00; H 04 L 9/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WPI

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 96/03 835 A2 (MACROVISION CORP.), 08 February 1996 (08.02.96), fig.1-3; abstract; claim 1.	1-5,9-12
A	EP 0 779 602 A2 (AT & T CORP.), 18 June 1997 (18.06.97), fig.1,4; abstract; claim 1.	1-16
A	US 5 598 473 A (LINSKER et al.), 28 January 1997 (28.01.97), abstract; fig.1,2; claim 1.	1-16

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:
„A" document defining the general state of the art which is not considered to be of particular relevance
„E" earlier application or patent but published on or after the international filing date
„L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
„O" document referring to an oral disclosure, use, exhibition or other means
„P" document published prior to the international filing date but later than the priority date claimed

„T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
„X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
„Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
„&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
25 January 1999 (25.01.99)	01 September 1999 (01.09.99)
Name and mailing adress of the ISA/AT Austrian Patent Office Kohlmarkt 8-10; A-1014 Vienna Facsimile No. 1/53424/200	Authorized officer Berger Telephone No. 1/53424/453

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SG 98/00104

Box I **Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)**

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. [X] Claims Nos.: 17 to 24
because they relate to subject matter not required to be searched by this Authority, namely:
the features of these claims 17 to 24 disclose a computer program.

2. [ ] Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. [ ] Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II **Observations where unity of invention is lacking (Continuation of item 2 of first sheet)**

This International Searching Authority found multiple inventions in this international application, as follows:

1. [ ] As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. [ ] As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. [ ] As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. [ ] No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

[ ] The additional search fees were accompanied by the applicant's protest.

[ ] No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/SG 98/00104

Im Recherchenbericht angeführtes Patentdokument Patent document cited in search report Document de brevet cité dans le rapport de recherche	Datum der Veröffentlichung Publication date Date de publication	Mitglied(er) der Patentfamilie Patent family member(s) Membre(s) de la famille de brevets	Datum der Veröffentlichung Publication date Date de publication
WO A2 9603835	08-02-96	AU A1 31276/95	22-02-96
		AU B2 697727	15-10-98
		BR A 9508340	09-09-97
		CA AA 2195939	08-02-96
		CN A 1159272	10-09-97
		EP A2 775418	28-05-97
		JP T2 10503338	24-03-98
		NZ A 290521	28-10-98
		US A 5574767	12-11-96
		WO A3 9603835	04-04-96
EP A2 779602	18-06-97	CA AA 2188975	16-06-97
		JP A2 9179583	11-07-97
		US A 5761329	02-06-98
US A 5598473	28-01-97	US A 5680455	21-10-97

Form PCT/ISA/210 (patent family annex) (July 1998)

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 :
G09C 5/00, G06F 17/00, G06K 7/00, 9/00, 9/36, 19/06, H04L 9/00

A1

(11) International Publication Number: **WO 00/70585**

(43) International Publication Date: 23 November 2000 (23.11.00)

(21) International Application Number: PCT/US00/13333

(22) International Filing Date: 15 May 2000 (15.05.00)

(30) Priority Data:

09/314,648	19 May 1999 (19.05.99)	US
09/342,688	29 June 1999 (29.06.99)	US
09/342,689	29 June 1999 (29.06.99)	US
09/342,971	29 June 1999 (29.06.99)	US
09/343,101	29 June 1999 (29.06.99)	US
09/343,104	29 June 1999 (29.06.99)	US
60/141,468	29 June 1999 (29.06.99)	US
60/151,586	30 August 1999 (30.08.99)	US
60/158,015	6 October 1999 (06.10.99)	US
60/163,332	3 November 1999 (03.11.99)	US
60/164,619	10 November 1999 (10.11.99)	US
09/531,076	18 March 2000 (18.03.00)	US
09/543,125	5 April 2000 (05.04.00)	US
09/547,664	12 April 2000 (12.04.00)	US
09/552,998	19 April 2000 (19.04.00)	US

(71) Applicant *(for all designated States except US)*: DIGIMARC CORPORATION [US/US]; Suite 250, 19801 SW 72nd Avenue, Tualatin, OR 97062 (US).

(72) Inventors; and

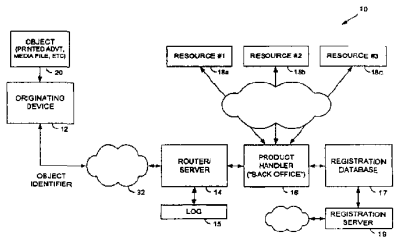
(75) Inventors/Applicants *(for US only)*: RHOADS, Geoffrey, B. [–/US]; 304 SW Tualatin Loop, West Linn, OR 97068 (US). RODRIGUEZ, Tony, F. [–/US]; 3104 NE 31st Avenue, Portland, OR 97212 (US). DAVIS, Bruce, L. [–/US]; 15599 Village Drive, Lake Oswego, OR 97034 (US). CARR, J., Scott [–/US]; 7814 SW 189th Avenue, Beaverton, OR 97007 (US). GROSSI, Brian, J. [–/US]; 220 Sleeper Avenue, Mountain View, CA 94040 (US). MCKINLEY, Tyler, J. [–/US]; 17020 SW Tracy Avenue, Lake Oswego, OR 97035 (US). SEDER, Phillip, A. [–/US]; 1600 Palatine Street, Portland, OR 97219 (US). PERRY, Burt, W. [–/US]; 15344 Provincial Hill Way, Lake Oswego, OR 97035 (US). HEIN, William, C., III [–/US]; 151 Indiantown Road, Glenmoore, PA 19343–1412 (US). MACINTOSH, Brian, T. [–/US]; 1200 Fairway Road, Lake Oswego, OR 97034 (US).

(74) Agent: CONWELL, William, Y.; Digimarc Corporation, 19801 SW 72nd Avenue, Suite 250, Tualatin, OR 97062 (US).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: METHODS AND SYSTEMS FOR CONTROLLING COMPUTERS OR LINKING TO INTERNET RESOURCES FROM PHYSICAL AND ELECTRONIC OBJECTS



(57) Abstract

Physical or electronic objects are encoded with identifiers, which serve to trigger object–appropriate responses from computer systems that encounter such objects. The encoding may be steganographic (e.g., by digital watermarks), so the presence of such identifiers is not evident to persons encountering the objects. An exemplary application is a computer system that looks at a printed magazine advertisement (20) and initiates a link to a corresponding internet page. In one such implementation, the computer system senses an identifer encoded in the advertisement, forwards the identifier to a remote database, receives from the database (17) a corresponding internet address (18a, 18b, 18c), and directs a browser to that address (18a, 18b, 18c). The same arrangement can be used for on–line ordering from printed merchandise catalogs. Another application is a computer system that looks at a printed spreadsheet (20), and retrieves from disk storage an electonic version of the same document for editing.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

METHODS AND SYSTEMS FOR CONTROLLING COMPUTERS OR LINKING TO INTERNET RESOURCES FROM PHYSICAL AND ELECTRONIC OBJECTS

Background and Summary of the Invention

"Bedoop." That might be the sound someone might hear as they lazily place a magazine advertisement in front of their desktop camera. Magically, the marketing and sales web site associated with the ad is displayed on their computer. More information? Want to buy now? Look at the full product line? No problem.

"Bedoop." That might be the same sound when that same someone places their credit card in front of their desktop camera. Instantly, the product displayed on the web page is purchased. Behind the scenes, a secure purchase link is initiated, transmitting all requisite information to the vendor. Twist the credit card clockwise and the purchaser chooses overnight delivery.

So goes an exemplary embodiment further described in this specification. Although this example is rather specific, it nevertheless alludes to vast array of applications possible when an input device such as a digital camera is turned into a general-purpose user interface device with an intuitive power that very well might rival the mouse and the keyboard.

One aspect of certain embodiments is that an object or paper product so-scanned contains digital information that can be quickly read and acted upon by an appropriately configured device, computer or appliance. Such an embodiment envisions that this digital information is aesthetically hidden on objects. These objects have been previously and pro-actively marked with the digital information, using any of the broad range of data encoding technologies, such as digital watermarking.

Although this aspect of the technology concentrates on flat object applications wherein the digital information is often imperceptibly integrated into the object, it is certainly not meant to be so limited. Objects can be three dimensional in nature and the information more visually overt and/or pre-existing (i.e., not "pro-actively" embedded, or it might not even be "digital," per se). Different implementation considerations attach to these variants. Likewise, though the bulk of this disclosure concentrates on objects which have some form of digital message attached thereto, some aspects of the technology may apply to objects which have no such thing, where the prior arts of pattern recognition and gestural input can be borrowed in combination with this technology to effect yet a broader array of applications.

Nor, as will be apparent, is the technology limited to systems employing optical input and encoded imagery. Corresponding techniques can also be employed with encoded audio. Indeed, any physical or electronic "object" can make use of the principles detailed herein.

"Bedoop." The sound that a refrigerator might make, outfitted with a simple camera/ processor unit/net connection, as a ten year old child holds up an empty milk carton and a ping goes out to the local grocery store, adding the item to an accumulating delivery list. The sound that might be heard echoing over and over inside Internet cafés as heretofore computerphobes take their first skeptical steps onto the World Wide Web. The sound heard at the fast food counter as a repeat customer holds up his sandwich

card ticking off his latest meal, hoping for the sirens to go off for a $500 prize given to the lucky customer of the week. Blue sky scenarios abound.

Such aspects of the present technology are thus about powerful new user interfaces to computers. These new user interfaces extend into the everyday world in ways that a mouse and keyboard never could. By enabling everyday objects to communicate their identities and functions to ever-attendant devices, not only will the World Wide Web be given an entirely new dimension, but basic home and office computing may be in store for some fundamental advances as well.

According to one aspect, the invention includes a method of data processing on a computer system, comprising (a) using an application program to compose an electronic version of a document; (b) printing the document onto paper, the printing including marking with machine readable indicia encoding plural-bit auxiliary data; and (c) storing the plural-bit auxiliary data in association with data identifying a location at which the electronic version of the document is stored.

According to another aspect, the invention includes a method of data processing on a computer system, comprising (a) presenting a printed document to an optical capture device; (b) processing image data produced by said device to decode plural-bit data encoded therein; (c) based on said decoded plural-bit data, launching a software application corresponding to said printed document; and (d) using said software application to open an electronic version of said document.

According to another aspect, the invention includes a method of operating a computer, the computer including an operating system with a registry database, the registry database associating specific data types with specific software programs particularly corresponding thereto, wherein the method further includes: (a) providing a frame of image data; (b) decoding plural-bit identifier data from the image data; (c) consulting the registry database to identify a software program corresponding to said identifier data; and (d) invoking the identified software program.

According to another aspect, the invention includes a greeting card having a substrate with visually-perceptible indicia printed thereon, wherein the card is encoded with plural-bit binary data that can be decoded by an image processing device and used to direct a computer to a web site where an image, video, and/or audio presentation corresponding to said card is provided.

According to another aspect, the invention includes a method of providing a customized greeting, comprising: (a) providing a greeting card having plural-bit data encoded therein; (b) customizing a web site presentation corresponding to said card; (c) providing the card to a recipient; (d) decoding the encoded plural-bit data from the card; and (e) in response to the decoded plural-bit data, presenting to the recipient the web site presentation.

According to another aspect, the invention includes a method of printing a magazine, comprising: (a) processing an electronic representation of an advertisement to encode plural bit data therein; (b) printing a page of advertising in accordance with said electronic representation to yield an encoded advertisement page; and (c) binding said page into a magazine; wherein said plural bit data serves to identify an entry in a database, said database entry having an internet address of a web page that is associated with said advertisement stored therein.

According to another aspect, the invention includes a promotional method comprising: (a) encoding a print advertisement to hide plural-bit data therein; (b) processing the print advertisement to extract the plural-bit data therefrom; and (c) using at least a part of the extracted plural-bit data to direct an internet web browser to a web site that provides consumer information related to a product or service promoted by the print advertisement.

According to another aspect, the invention includes a method of determining consumer response to print advertising, comprising: (a) encoding a first print advertisement with first data; (b) encoding a second print advertisement with second data different than the first; (c) the first and second data providing identifiers by which consumer devices can link to web pages associated with said advertisements; (d) monitoring linking traffic due to each of said identifiers to thereby determine consumer response to the advertisements.

According to another aspect, the invention includes a promotional method comprising: (a) presenting an object within the field of view of an optical sensor device, the object being selected from the list consisting of a retail product, packaging for a retail product, or printed advertising; (b) acquiring optical data corresponding to the object; (c) decoding plural-bit digital data from the optical data; (d) submitting at least some of said decoded data to a remote computer; and (e) determining at the remote computer whether a prize should be awarded in response to submission of said decoded data.

According to another aspect, the invention includes a method of interacting with a magazine using a computer, the computer including an internet web browser, the method including: (a) providing a peripheral device having a sensor; (b) positioning the peripheral device adjacent a first advertisement in the magazine to direct the web browser to a first internet address; and (c) positioning the peripheral device adjacent a second advertisement in the magazine to direct the web browser to a second internet address.

According to another aspect, the invention includes a computer peripheral and method of its use, the peripheral being used in conjunction with a computer system having an internet browser associated therewith, the peripheral comprising: (a) a housing adapted to fit within a user's palm and slide over a medium; (b) an optical sensor having at least one sensing element and producing optical data; (c) a lens for imaging the medium onto the sensor; and the method includes: (a) sliding the peripheral over a portion of a printed advertisement; (b) processing the optical data to decode plural bit information encoded on the advertisement; and (c) using said plural bit information to direct the internet browser to an internet web page associated with said advertisement.

According to another aspect, the invention includes an electronic commerce method comprising: (a) providing a printed catalog that includes an image of an article offered for sale by a merchant, wherein the image is encoded with plural-bit binary data; (b) optically sensing the image to produce optical data corresponding thereto; (c) decoding the encoded data from the optical data; and (d) electronically ordering the article from the merchant by use of said decoded data. The ordering may make use of earlier-stored customer profile information (e.g., clothing size data), and the encoding may be steganographic.

According to another aspect, the invention comprises a wireless telephony handset including a microphone, a modulator, and an RF amplifier, the device serving to receive audio and transmit an RF signal conveying audio modulation, the handset further including an optical sensor producing optical data, a lens for imaging an object onto the sensor, and a decoder for decoding plural bit identifier data conveyed by a barcode or a digital watermark on the object.

According to another aspect, the invention includes an image-based network navigation method permitting a user to link to a remote computer, comprising: (a) detecting encoded data from a printed object; (b) linking to the remote computer through a network in accordance with said encoded data; and (c) providing the user's zip code to the remote computer.

According to another aspect, the invention includes a method comprising: (a) sensing an object identifier from a first object; (b) sending said first object identifier from a first device to a second device; (c) in response, at said second device, identifying address information corresponding to said first object identifier and sending same to the first device; (d) initiating a link from the first device in accordance with said address information; (e) at said second device, identifying additional objects related to said first object; identifying additional address information corresponding to said additional objects; and sending said additional address information to the first device; and (f) storing said additional address information in a memory at the first device; wherein, if an object included among said identified additional objects is sensed by the first device, the corresponding address information can be retrieved from said memory in the first device without the intervening delays of communicating with the second device.

According to another aspect, the invention includes an apparatus having a detector of machine readable data and a software program used in conjunction with said machine readable data, operable to transmit a packet of data to a remote system, said packet of data comprising (a) an identifier of said software program, and (b) at least a portion of detected machine readable data.

According to another aspect, the invention includes an apparatus having a detector of machine readable data and a software program used in conjunction with said machine readable data, operable to transmit a packet of data to a remote system, said packet of data comprising (a) a context or environment identifier, and (b) at least a portion of detected machine readable data.

According to another aspect, the invention includes a networked computer system, responsive to watermark data sent from a software program on a remote computer, to initiate delivery of advertisement data to said remote computer.

In any of the foregoing arrangements, the encoding can be steganographic (e.g., by digital watermarking), or can employ other machine readable data (e.g., barcodes, etc.). More generally, the arrangements just reviewed generally have counterparts that can be implemented with other than optical or image data (e.g., audio data, magnetic stripe information, etc.).

The foregoing just touches on a few of the many inventive aspects of the technology detailed below. These and other features of the present technology will be more readily apparent from the following detailed description, which proceeds with reference to the accompanying drawings.

Brief Description of the Drawings

Fig. 1 is a diagram showing the principal process components of an illustrative system employing the present technology.

Fig. 2 is a block diagram showing an illustrative system for performing the response process of Fig. 1.

Fig. 3 is a block diagram more particularly detailing an originating device used in the system of Fig. 2.

Fig. 4 illustrates certain top level data flows in the system of Fig. 2.

Fig. 5 illustrates certain data flows associated with the router of Fig. 2.

Fig. 6 illustrates certain data flows associated with the registration process of Fig. 2.

Fig. 7 illustrates certain data flows associated with the product handler of Fig. 2.

Figs. 8-10 show a sequence of screen shots from an illustrative system.

Fig. 11 is a block diagram showing another embodiment of the present technology.

Fig. 12 is a block diagram showing another embodiment of the present technology.

Fig. 13 is a block diagram of a prior art scanner.

Fig. 14 shows an object being scanned along an arced path.

Fig. 15 shows how object warping may be detected.

Fig. 16 shows use of binocular processing to determine certain object attributes.

Fig. 17 shows a watermark grid.

Fig. 18 shows a pre-warped watermark grid.

Fig. 19 shows another pre-warped watermark grid.

Detailed Description

Basically, the technology detailed in this disclosure may be regarded as enhanced systems by which users can interact with computer-based devices. Their simple nature, and adaptability for use with everyday objects (e.g., milk cartons), makes the disclosed technology well suited for countless applications.

Due to the great range and variety of subject matter detailed in this disclosure, an orderly presentation is difficult to achieve. For want of a better arrangement, the specification is broken into two main parts. The first part details a variety of methods, applications, and systems, to illustrate the diversity of the present technology. The second more particularly focuses on a print-to-internet application. A short concluding portion is presented in Part III.

As will be evident, many of the topical sections presented below are both founded on, and foundational to, other sections. For want of a better rationale, the sections in the first part are presented in a more or less random order. It should be recognized that both the general principles and the particular details from each section find application in other sections as well.

Compounding the situation, the present specification draws from several priority applications filed over the course of nearly a year. Accordingly, the same concepts are sometimes expressed several

times, each reflecting a different perspective – depending on the date and context of the application in which it was first found.

The term "Bedoop" has been superceded in applicant's commercialization of the technology by the term Digimarc MediaBridge. Both terms are used in this specification but refer to the same technology.

To prevent the length of this disclosure from compounding out of control, the various permutations and combinations of the features of the different sections are not exhaustively detailed. The inventors intend to explicitly teach such combinations/permutations, but practicality requires that the detailed synthesis be left to those who ultimately implement systems in accordance with such teachings.

PART I

Introduction to Digital Watermarking and other Encoding Techniques

There are nearly as many techniques for digital watermarking (a type of steganographic data encoding) as there are applications for it. The reader is presumed to be familiar with the great variety of methods. A few are reviewed below.

The present assignee's prior application 09/127,502, filed July 31, 1998, now published as WO0007356, shows techniques by which very fine lines can be printed on a medium to slightly change the medium's apparent tint, while also conveying digital data. Commonly-owned application 09/074,034, filed May 6, 1998, now published as WO 9953428, details how the contours of printed imagery can be adjusted to convey digital data. (That technique can be applied to printed text characters, as well as the line art imagery particularly considered.) The assignee's patent 5,850,481 details how the surface of paper or other media can be textured to convey optically-detectable binary data. The assignee's patents 5,862,260, 5,841,886 and 5,809,160 detail various techniques for steganographically encoding photographs and other imagery.

Some watermarking techniques are based on changes made in the spatial domain; others are based on changes made in transformed domains (e.g., DCT, wavelet).

Watermarking of printed text can be achieved by slight variations to character shape, character kerning, line spacing, etc., as shown by various writings by Brassil et al, including "Electronic Marking and Identification Techniques to Discourage Document Copying," Proceedings of INFOCOM '94 Conference on Computer, IEEE Comm. Soc Conference, June 12-16, 1994, pp. 1278-1287; "Hiding Information in Document Images," Proceedings of the Twenty-Ninth Annual Conference on Information Sciences and Systems, p. 482-9, 1995; and "Document marking and identification using both line and word shifting," Proceedings of IEEE INFOCOM '95, The Conference on Computer Communications. Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies; Bringing Information to People (Cat. No.95CH35759), p. 853-60 vol.2, 1995.

The foregoing is just a sampling of the large literature on watermarking. The artisan is presumed to be familiar with such art, all of which is generally suitable for use with the novel concepts detailed below.

Although the following specification focuses on applications employing digital watermarking, certain of such applications can alternatively employ other data encoding techniques, including 1D and 2D barcodes, magnetic ink character recognition (MICR), optical character recognition (OCR), optical mark recognition (OMR), radio frequency identification (RF/ID), UV/IR identification technologies, data glyphs, organic transistors, magnetic stripe, etc., depending on the particular application requirements.

Basic Principles – Refrigerators and Desktop Clutter

Referring to Fig. 11, a basic embodiment 110 of the present technology includes an optical sensor 112, a computer 114, and a network connection 116 to the internet 118. The illustrated optical sensor 112 is a digital camera having a resolution of 320 by 200 pixels (color or black and white) that stares out, grabbing frames of image data five times per second and storing same in one or more frame buffers. These frames of image data are analyzed by a computer 114 for the presence of Bedoop data. (Essentially, Bedoop data is any form of plural-bit data encoding recognized by the system 110 -- data which, in many embodiments, initiates some action.) Once detected, the system responds in accordance with the detected Bedoop data (e.g., by initiating some local action, or by communication with a remote computer, such as over the internet, via an online service such as AOL, or using point-to-point dial-up communications, as with a bulletin board system).

Consider the milk carton example. The artwork on a milk carton can be adapted to convey Bedoop data. In the preferred embodiment, the Bedoop data is steganographically encoded (e.g., digitally watermarked) on the carton. Numerous digital watermarking techniques are known – all of which convey data in a hidden form (i.e., on human inspection, it is not apparent that digitally encoded data is present). Exemplary techniques operate by slightly changing the luminance, or contours of selected points on artwork or text printed on the carton, or by splattering tiny droplets of ink on the carton in a seemingly random pattern. Each of these techniques has the effect of changing the local luminance at areas across the carton – luminance changes that can be detected by the computer 114 and decoded to extract the encoded digital data. In the case of a milk carton, the data may serve to identify the object as, e.g., a half gallon carton of Alpenrose brand skim milk.

The Fig. 11 apparatus can be integrated into the door of a refrigerator and used to compile a shopping list. Milk cartons, and other Bedoop-encoded packaging, can be held up the optical sensor. When the computer 114 detects the presence of Bedoop data and successfully decodes same, it issues a confirmation tone ("be-doop") from a speaker or other audio transducer 122. The computer then adds data identifying the just-detected object to a grocery list. This list can be maintained locally (in disk storage or non-volatile RAM 124 or the like in the refrigerator or elsewhere in the home), or remotely (e.g., at a server computer located at a user-selected grocery, or elsewhere). In either event, the list is desirably displayed on a display in the user's home (e.g., an LCD screen 126 built into the front of the appliance). Conventional user interface techniques can be employed permitting the user to scroll through the displayed list, delete items as desired, etc.

Periodically, the listed groceries can be purchased and the list cleared. In one embodiment, the list is printed (either at the home or at the grocery), and the user walks the grocery aisles and purchases same in the conventional manner. In another embodiment, the grocer pulls the listed items from the shelves (in response to a user request conveyed by the internet or telephone, or by a gesture as hereafter detailed). Once the list has been pulled, the grocer can alert the user that the groceries are available for pickup (again, e.g., by internet or telephone message), or the grocer can simply deliver the groceries directly to the user's home. Naturally, on-line payment mechanisms can be employed if desired.

Consider a wholly unrelated Bedoop application. A Microsoft Excel spreadsheet is printed onto paper, and the paper becomes buried in a stack of clutter on an office worker's desk. Months later the spreadsheet again becomes relevant and is dug out of the stack. Changes need to be made to the data, but the file name has long-since been forgotten. The worker simply holds the printed page in front of a camera associated with the desktop computer. A moment later, the electronic version of the file appears on the worker's computer display.

When the page was originally printed, tiny droplets of ink or toner were distributed across the paper in a pattern so light as to be essentially un-noticeable, but which steganographically encoded the page with a plural-bit binary number (e.g., 24-128 bits). A database (e.g., maintained by the operating system, the Excel program, the printer driver, etc.) stored part of this number (e.g., 20 bits, termed a Universal Identifier, or UID) in association with the path and file name at which the electronic version of the file was stored, the page number within the document, and other useful information (e.g., author of the file, creation date, etc.).

The steganographic encoding of the document, and the updating of the database, can be performed by the software application (e.g., Excel). This option can be selected once by the user and applied thereafter to all printed documents (e.g., by a user selection on an "Options" drop-down menu), or can be presented to the user as part of the Print dialog window and selected (or not) for each print job.

When such a printed page is later presented to the camera, the computer automatically detects the presence of the encoded data on the page, decodes same, consults the database to identify the file name/location/page corresponding to the 20-bit UID data, and opens the identified file to the correct page (e.g., after launching Excel). This application is one of many "paper as portal" applications of the Bedoop technology.

The foregoing are but two of myriad applications of the technology detailed herein. In the following discussion a great many other applications are disclosed. However, regardless of the length of the specification, it is possible only to begin to explore a few of the vast ramifications of this technology.

A few more details on the basic embodiments described above may be helpful before delving into other applications.

Optics

For any system to decode steganographically encoded data from an object, the image of the object must be adequately focused on the digital camera's CCD (or other, e.g., CMOS) sensor. In a low

cost embodiment, the camera has a fixed nominal focal length, e.g., in the range of 2-24 inches (greater or lesser lengths can of course be used). Since the camera is continuously grabbing and analyzing frames of data, the user can move the object towards- or away- from the sensor until the decoder succeeds in decoding the steganographically encoded data and issues a confirming "Bedoop" audio signal.

In more elaborate embodiments, known auto-focusing technology can be employed.

In still other embodiments, the camera (or other sensor) can be equipped with one or more auxiliary fixed-focus lenses that can be selectively used, depending on the particular application. Some such embodiments have a first fixed focused lens that always overlies the sensor, with which one or more auxiliary lenses can be optically cascaded (e.g., by hinge or slide arrangements). Such arrangements are desirable, e.g., when a camera is not a dedicated Bedoop sensor but also performs other imaging tasks. When the camera is to be used for Bedoop, the auxiliary lens is positioned (e.g., flipped into) place, changing the focal length of the first lens (which may by unsuitably long for Bedoop purposes, such as infinity) to an appropriate Bedoop imaging range (such as one foot).

Other lens-switching embodiments do not employ a fixed lens that always overlies the sensor, but instead employ two or more lenses that can be moved into place over the sensor. By selecting different lenses, focal lengths such as infinity, six feet, and one foot can be selected.

In all such arrangements, it is desirable (but not essential) that the steganographically-encoded portion of the object being imaged fills a substantial part of the image frame. The object can be of various sizes, e.g., a 10 by 12 inch front panel of a cereal box, or a proof of purchase certificate that is just one inch square. To meet this requirement, small objects need to be placed closer to the camera than large objects. The optics of the system can be designed, e.g., by selection of suitable aperture sizing and auxiliary lighting (if needed), to properly image objects of various sizes within a range of focal distances.

Some embodiments avoid issues of focal distances and identifying the intended object by constraining the size of the object and/or its placement. An example is a business card reader that is designed for the sole task of imaging business cards. Various such devices are known.

Decoding/Encoding

The analysis of the image data can be accomplished in various known ways. Presently, most steganographic decoding relies on general purpose microprocessors that are programmed by suitable software instructions to perform the necessary analysis. Other arrangements, such as using dedicated hardware, reprogrammable gate arrays, or other techniques, can of course be used.

The steganographic decoding process may entail three steps. In the first, the object is located. In the second, the object's orientation is discerned. In the third, the Bedoop data is extracted from the image data corresponding to the Bedoop object.

The first step, object location, can be assisted by various clues. One is the placement of the object; typically the center of the image field will be a point on the object. The surrounding data can then be analyzed to try and discern the object's boundaries.

Another location technique is slight movement. Although the user will typically try to hold the object still, there will usually be some jitter of the Bedoop object within the image frame (e.g., a few pixels back and forth). Background visual clutter, in contrast, will typically be stationary. Such movement may thus be sensed and used to identify the Bedoop object from within the image data.

Still another object-location clue is object shape. Many Bedoop objects are rectangular in shape (or trapezoidal as viewed by the camera). Straight edge boundaries can thus be used to define an area of likely Bedoop data.

Color is a further object identification clue that may be useful in some contexts.

Yet another object location clue is spatial frequency. In imaging systems with well defined focal zones, undesired visual clutter may be at focal distances that results in blurring. The Bedoop object, in contrast, will be in focus and may be characterized by fine detail. Analyzing the image data for the high frequencies associated with fine detail can be used to distinguish the intended object from others.

Characteristic markings on the object (as discussed below in connection with determining object orientation) can also be sensed and used in locating the object.

Once the Bedoop object has been located within the image data, masking can be applied (if desired) to eliminate image data not corresponding to the intended object.

The next step in the decoding process, determining orientation of the Bedoop data, can likewise be discerned by reference to visual clues. For example, some objects include subliminal graticule data, or other calibration data, steganographically encoded with the Bedoop data to aid in determining orientation. Others can employ overt markings, either placed for that sole purpose (e.g. reference lines or fiducials), or serving another purpose as well (e.g. lines of text), to discern orientation. Edge-detection algorithms can also be employed to deduce the orientation of the object by reference to its edges.

Some embodiments filter the image data at some point in the process to aid in ultimate Bedoop data extraction. One use of such filtering is to mitigate image data artifacts due to the particular optical sensor. For example, CCD arrays have regularly-spaced sensors that sample the optical image at uniformly spaced discrete points. This discrete sampling effects a transformation of the image data, leading to certain image artifacts. An appropriately configured filter can reduce some of these artifacts.

(In some arrangements, the step of determining the orientation can be omitted. Business card readers, for example, produce data that is reliably free of artifacts and is of known scale. Or the encoding of the Bedoop data can be effected in such a way that renders it relatively immune to certain distortion mechanisms. For example, while the presently-preferred encoding arrangement operates on a 2D grid basis, with rows and columns of data points, the encoding can alternatively be done on another basis (e.g., a rotationally-symmetric form of encoding, so that rotational state of the image data can be ignored). In still other embodiments, the orientation-determining step can be omitted because the decoding can readily proceed without this information. For example decoding which relies on the Fourier-Mellin transform produces data in which scale and rotation can be ignored.)

Once the orientation of the object is discerned, the image data may be virtually re-registered, effectively mapping it to another perspective (e.g., onto a rectilinear image plane). This mapping can employ known image processing techniques to compensate, e.g., for rotation state, scale state, differential scale state, and X-Y offset, of the original Bedoop image data. The resulting frame of data may then be more readily processed to extract the steganographically-encoded Bedoop data.

In an exemplary embodiment, after the image data is remapped into rectilinear planar form, subliminal graticule data is sensed that identifies the locations within the image data where the binary data is encoded. Desirably, the binary data is redundantly encoded, e.g., in blocks of eight-by-eight patches. Each patch comprises one or more pixels. (The patches are typically square, and thus contain 1, 4, 9, or 16, etc. pixels.) The nominal luminance of each patch before encoding (e.g., artwork pre-existing on the object) is slightly increased or decreased to encode a binary "1" or "0." The change is slight enough to be generally imperceptible to human observers, yet statistically detectable from the image data – especially if several such blocks are available for analysis. Preferably, the degree of change is adapted to the character of the underlying image, with relatively greater changes being made in regions where the human eye is less likely to notice them. Each block thus encoded can convey plural bits of data (e.g., 16 – 128 bits). The encoding of such blocks in tiled fashion across the object permits the data to be conveyed in robust fashion.

Much of the time, of course, the Bedoop sensor is staring out and grabbing image frames that have no Bedoop data. Desirably, the detection process includes one or more checks to assure that Bedoop data is not wrongly discerned from non-Bedoop image data. Various techniques can be employed to validate the decoded data, e.g., error detecting codes can be included in the Bedoop payload and checked to confirm correspondence with the other Bedoop payload. Likewise, the system can confirm that the same Bedoop data is present in different tiled excerpts within the image data, etc.

Details of particular encoding and decoding techniques can be found in U.S. patent 5,862,260 and application 09/503,881. As noted, the data can be encoded on a tiled basis, with each tile being 64 to 256 elements on a side. Each element can be 0.01 inches square. The Bedoop payload data can be redundantly represented by various error-tolerant coding techniques (e.g., convolutions coding, trellis coding, turbo codes, etc.) to fill the tiled block. Each bit is thus redundantly encoded, with a "1" being represented as an increase at certain pixels, and as a decrease at other pixels. The increases and decreases can be scaled in accordance with visual masking attributes of the image being encoded. The calibration signal can be summed with the tiled data signal and can comprise a signal tailored in the frequency domain to have 12 – 64 spectral impulses per quadrant, in a known pattern. During detection, the rotation or scaling of these impulses from their known frequency domain coordinates permits the rotation or scaling of the image to be discerned and compensated for.

Data Structures, Formats, Protocols, and Infrastructures

In an exemplary system, the Bedoop data payload is 64 bits. This payload is divided into three fields CLASS (12 bits), DNS (24 bits) and UID (24 bits). (Other payload lengths, fields, and divisions, are of course possible, as is the provision of error-checking or error-correcting bits.)

Briefly, the CLASS ID is the most rudimentary division of Bedoop data, and may be analogized, in the familiar internet taxonomy, to the limited number of top level domains (e.g., .com, .net, .org, .mil, .edu, .jp, .de, .uk, etc.). It is basically an indicator of object type. The DNS ID is an intermediate level of data, and may be analogized to internet server addresses (e.g., biz.yahoo, interactive.wsj, etc.) The UID is the finest level of granularity, and can roughly be analogized to internet pages on a particular server (e.g., edition/current/summaries/front.htm, daily/home/default.hts, etc.).

Generally speaking, the CLASS ID and DNS ID, collectively, indicate to the system what sort of Bedoop data is on the object. In the case of Bedoop systems that rely on remote servers, the CLASS and DNS IDs are used in identifying the server computer that will respond to the Bedoop data. The UID determines precisely what response should be provided.

In the case of a refrigerator Bedoop system, what happens if an object with an unfamiliar CLASS/DNS ID data is encountered? The system can be programmed not to respond at all, or to respond with a raspberry-like sound (or other feedback) indicating, "I see a Bedoop object but don't know what to do with it."

Most systems will be able to respond to several classes of Bedoop objects. Simple software-based systems can compare the CLASS/DNS ID (and optionally the UID) to fixed values, and can branch program execution to corresponding subroutines. Likewise, hardware-based systems can activate different circuitry depending on the detected CLASS/DNS ID.

In the case of a computer equipped with a Bedoop input device (e.g., a Sony VAIO PictureBook laptop with built-in camera, or a desktop personal computer with a tethered camera), the operating system's registry database can be employed to associate different application programs with different CLASS/DNS IDs (just as the .XLS and .DOC file extensions are commonly associated by existing operating system registries to invoke Microsoft Excel and Word software applications, respectively). When a new Bedoop application is installed, it logs an entry in the registry database indicating the CLASS/DNS ID(s) that it will handle. Thereafter, when an object with such a CLASS/DNS ID is encountered, the operating system automatically launches the corresponding application to service the Bedoop data in an appropriate manner.

Sometimes the computer system may encounter a Bedoop object for which it does not have a registered application program. In such case, a default Bedoop application can be invoked. This default application can, e.g., establish an internet link to a remote server computer (or a network of such computers), and can transmit the Bedoop data (or a part of the Bedoop data) to that remote computer. The remote server can undertake the response itself, it can instruct the originating computer how to respond appropriately, or it can undertake some combination of these two responses. (Such arrangements are further considered below.)

Fig. 12 shows an illustrative architecture employing the foregoing arrangement.

At a local Bedoop system 128 (which may be implemented, for example, using a conventional personal computer 129), a camera, scanner, or other optical sensor 130 provides image data to a decoder 132 (which may be implemented as a software component of the operating system 133). The decoder 132 analyzes the image data to discern the plural-bit Bedoop data. The CLASS ID of this Bedoop data is applied to a Bedoop registry 134. The registry responds by identifying and launching a local Bedoop application 136 designed to service the discerned Bedoop data.

Sometimes the system 128 may encounter a Bedoop object for which several different responses may be appropriate. In the case of a printed office document, for example, one response may be as described above – to present the electronic version of the file on a computer, ready for editing. But other responses may also be desired, such as writing an email message to the author of the printed document, with the author's email address already specified in the message address field, etc.

Such different responses may be handled by different Bedoop applications, or may be options that are both provided by a single Bedoop application. In the former case, when the CLASS/DNS IDs are decoded and provided to the operating system, the registry indicates that there are two (or more) programs that might be invoked. The operating system can then present a dialog box to the user inviting the user to specify which form of response is desired. Optionally, a default choice can be made if the user doesn't specify within a brief period (e.g., three seconds). The operating system can then launch the Bedoop application corresponding to the chosen response.

A similar arrangement can be employed if a single Bedoop application can provide both responses. In such case the operating system launches the single Bedoop application (since there is no ambiguity to be resolved), and the application presents the choice to the user. Again, the user can select, or a default choice can be automatically made.

In the just-described situations, the user can effect the choice by using the keyboard or mouse -- as with traditional dialog boxes. But Bedoop provides another, usually easier, form of interaction. The user can make the selection through the optical sensor input. For example, moving the object to the right can cause a UI button on the right side of the dialog box to be selected; moving the object to the left can cause a UI button on the left side of the dialog box to be selected; moving the object towards the camera can cause the selected button to be activated. Many other such techniques are possible, as discussed below.

If the registry 134 does not recognize, or otherwise does not know how to respond to Bedoop data of that particular CLASS/DNS, the registry launches a default Bedoop client application. This client application, in turn, directs a web browser 40 on the local Bedoop system 128 to communicate with a remote master registration server computer 42. The local computer forwards the Bedoop data to this master server. The master server 42 examines the CLASS ID, and forwards the Bedoop data (directly, or through intervening servers) to a corresponding CLASS server 44. (A single server may handle Bedoop data of several classes, but more typically there is a dedicated server for each CLASS.)

Each CLASS server 44 serves as the root of a tree 46 of distributed DNS servers. A DNS server 48a, for example, in a first tier 50 of the DNS server tree, may handle Bedoop data having DNS IDs beginning with "000." Likewise, DNS server 48b may handle Bedoop data having DNS IDs beginning with "001," etc., etc.

Each DNS server in the first tier 50 may, in turn, route Bedoop data to one of 8 servers in a second tier of the tree, in accordance with the fourth- through sixth bits of the DNS data. The tree continues in this fashion until a terminal level of DNS leaf node servers 56.

Ultimately, Bedoop data routed into this network reaches a DNS leaf node server 56. That leaf node server may handle the Bedoop data, or may redirect the local Bedoop system to a further server 58 that does so. That ultimate server – whether a DNS leaf node server or a further server – can query the local Bedoop system for further information, if necessary, and can either instruct the local Bedoop system how to respond, or can undertake some or all of the response itself and simply relay appropriate data back to the local Bedoop system.

In arrangements in which the local Bedoop system is redirected, by the DNS leaf node server, to a further server that actually handles the response, access to the further server may be through a port 59 (e.g., a special URL) tailored to receipt of Bedoop data.

In a typical implementation, most or all of the servers are mirrored, or otherwise replicated/redundant, so that failure of individual computers does not impair operation of the system.

Caching can be provided throughout the trees of servers to speed responses. That is, responses by leaf nodes for certainly commonly-encountered CLASS/DNS IDs can be temporarily stored earlier in the tree(s). Bedoop data, propagating through the server network, can prompt a response from an intermediate server if there is a cache hit.

If desired, Bedoop traffic through the above-detailed server trees can be monitored to collect demographic and statistical information as to what systems are sending what Bedoop data, etc. One use of such information is to dynamically reconfigure the DNS network to better balance server loads, to virtually relocate DNS resources nearer regions of heavy usage, etc. Another use of such information is for marketing purposes, e.g., to promote certain Bedoop features and applications within user groups (e.g., internet domains) that seem to under-utilize those features.

Within certain user networks that are linked to the internet, e.g., corporate networks, Bedoop data that isn't handled within the originating Bedoop system may first be routed to a Bedoop name server within the corporate network. That server will recognize certain types of Bedoop data, and know of resources within the corporate network suitable for handling same. Referral to such resources within the corporate network will be made, where possible. These resources (e.g., corporate servers) may respond to Bedoop data in a way customized to the corporate preferences. If the corporate Bedoop name server does not know of a resource within the corporate network that can respond to the Bedoop data, the corporate name server then routes the data to the public Bedoop network described above. (Such referral can be to the master registration server or, to the extent the corporate name server knows the addresses of

appropriate servers within the DNS server tree, or of the further servers to which DNS servers may point for certain Bedoop data, it can redirect the local Bedoop system accordingly.)

In typical rich Bedoop implementations, local systems may have libraries of Bedoop services, applications, or protocols. Some may be unique to that computer. Others may be commonly available on all computers. Some may be highly secure, employing encryption and/or anti-hacking measures, or data protocols that are not generally recognized. Others may be shareware, or the result of open-source programming efforts.

While the just-described arrangements used a 12/24/24 bit protocol for CLASS/DNS/UID data, other arrangements can of course be used. In some applications it is advantageous for the protocol to more nearly match those commonly used for internet communications. For example, IP addresses for internet Domain Name Servers (DNS) are presently 32 bits, with extension to 64 or 128 bits foreseen in the near future. The DNS field in Bedoop systems can be follow the internet standard.

Greeting Cards, Birthday Cards, Etc.

To further illustrate some of the basic principles associated with this technology, consider greeting cards and the like that are encoded (e.g., by texturing, printing, etc.) with Bedoop data. On receiving such a card, a recipient holds it in front of the image capture device on a laptop or other computer. The computer responds by displaying an internet web page that has a stock- or customized-presentation (image, video, audio-video, etc.) to complement that presented on the greeting card.

The web site presentation can be personalized by the sender (e.g., with a text message, recent family photographs, etc.), either at the point of card sale, or sometime after the card is purchased. In the latter case, for example, the card can be serialized. After taking the card home, the purchaser can visit the card vendor's web site and enter the card serial number in an appropriate user interface. The purchaser is then presented with a variety of simple editing tools to facilitate customization of the web greeting. When the sender is finished designing the web greeting, the finished web page data is stored (by software at the vendor's web site) at a site corresponding to the serial number.

When the card is received by a recipient and held in front of a Bedoop sensor, CLASS, DNS, and UID data is decoded from the card. The CLASS and DNS data are used to navigate the earlier-described server network to reach a corresponding DNS leaf node server (perhaps maintained by the Hallmark greeting card company). That leaf node server indexes a table, database, or other data structure with the UID from the Bedoop data, and obtains from that data structure the address of an ultimate web site – the same address at which the web greeting customized by the sender was stored. That address is provided by the DNS leaf node server back to the local computer, with instructions that the web page at that address be loaded and displayed (e.g., by HTML redirection). The local computer complies, presenting the customized web greeting to the card recipient.

In the just-described embodiment, in which a pre-encoded card is purchased by a sender and the web-display is then customized, the address of the web site is typically determined by the card vendor.

But this need not be the case. Likewise, the card need not be "purchased" in the typical, card-shop fashion.

To illustrate the foregoing alternatives, consider the on-line acquisition of a greeting card, e.g., by visiting a web site specializing in greeting cards. With suitable user-selection (and, optionally, customization), the desired card can be printed using an ink-jet or other printer at the sender's home. In such case, the Bedoop data on the card can be similarly customized. Instead of leading to a site determined by the card vendor, the data can lead to the sender's personal web page, or to another arbitrary web address.

To effect such an arrangement, the sender must arrange for a DNS leaf node server to respond to a particular set of Bedoop data by pointing to the desired web page. While individuals typically will not own DNS servers, internet service providers commonly will. Just as AOL provides simple tools permitting its subscribers to manage their own modest web pages, internet service providers can likewise provide simple tools permitting subscribers to make use of DNS leaf node servers. Each subscriber may be assigned up to 20 UIDs. The tools would permit the users to define a corresponding web address for each UID. Whenever a Bedoop application led to that DNS leaf node server, and presented one of those UIDs, the server would instruct the originating computer to load and present the web page at the corresponding web address.

Prior to customizing the greeting card, the sender uses the tool provided by the internet service provider to store the address of a desired destination web address in correspondence with one of the sender's available UIDs. When customizing the greeting card, the sender specifies the Bedoop data that is to be encoded, including the just-referenced UID. The greeting card application encodes this data into the artwork and prints the resulting card. When this card is later presented to a Bedoop system by the recipient, the recipient's system loads and displays the web page specified by the sender.

Commerce in Bedoop Resources

In the just-described arrangement, internet service providers make available to each subscriber a limited number of UIDs on a DNS server maintained by the service. Business enterprises typically need greater Bedoop resources, such as their own DNS IDs (or even their own CLASS ID(s).

While variants of the Bedoop system are extensible to provide an essentially unlimited number of CLASS IDs and DNS IDs, in the illustrated system these resources are limited. Public service, non-profit, and academic applications should have relatively generous access to Bedoop resources, either without charge or for only a modest charge. Business enterprises, in contrast, would be expected to pay fees to moderate their potentially insatiable demand for the resources. Small businesses could lease blocks of UIDs under a given CLASS/DNS ID. Larger businesses could acquire rights to entire DNS IDs, or to entire CLASS IDs (at commensurately greater fees).

Web-based systems for assigning DNS IDs (and CLASS IDs) can be modeled after those successfully used by Internic.com, and now Networksolutions.com, for registration of internet domains. The user fills out a web-based form with names, addresses, and billing information; the system makes the

necessary changes to all of the hidden system infrastructure – updating databases, routing tables, etc., in servers around the world.

Controlled-Access ID

Just as the above-described embodiment employed an ink-jet printer to produce a customized-Bedoop greeting card, the same principles can likewise be applied to access-control objects, such as photo-IDs.

Consider an employment candidate who will be interviewing at a new employer. The candidate's visit is expected, but she is not recognized by the building's security personnel. In this, and many other applications, arrangements like the following can be used:

The employer e-mails or otherwise sends the candidate an access code. (The code can be encrypted for transmission.) The code is valid only for a certain time period on a given date (e.g., 9:00 a.m. – 11:00 a.m. on June 28, 1999).

Upon receipt of the access code, the candidate downloads from the web site of the state Department of Motor Vehicles the latest copy of her driver's license photo. The DMV has already encoded this photo with Bedoop data. This data leads to a state-run DNS leaf node server 56. When that server is presented with a UID decoded from a photograph, the server accesses a database and returns to the inquiring computer a text string indicating the name of the person depicted by the photograph.

The candidate incorporates this photo into an access badge. Using a software application (which may be provided especially for such purposes, e.g., as part of an office productivity suite such as Microsoft Office), the photo is dragged into an access badge template. The access code emailed from the employer is also provided to this application. On selecting "Print," an ink-jet printer associated with the candidate's computer prints out an access badge that includes her DMV photo and her name, and is also steganographically encoded in accordance with the employer-provided access code.

The name printed on the badge is obtained (by the candidate's computer) from the DMV's DNS server, in response to Bedoop data extracted from the photograph. (In this application, unlike most, the photograph is not scanned as part of a Bedoop process. Instead, the photograph is already available in digital form, so the Bedoop decoding proceeds directly from the digital representation.)

For security purposes, the access code is not embedded using standard Bedoop techniques. Instead, a non-standard format (typically steganographic) is employed. The embedding of this access code can span the entire face of the card, or can be limited to certain regions (e.g., excluding the region occupied by the photograph).

On the appointed day the candidate presents herself at the employer's building. At the exterior door lock, the candidate presents the badge to an optical sensor device, which reads the embedded building access code, checks it for authenticity and, if the candidate arrived within the permitted hours, unlocks the door.

Inside the building the candidate may encounter a security guard. Seeing an unfamiliar person, the guard may visually compare the photo on the badge with the candidate's face. Additionally, the

guard can present the badge to a portable Bedoop device, or to one of many Bedoop systems scattered through the building (e.g., at every telephone). The Bedoop system extracts the Bedoop data from the card (i.e., from the DMV photograph), interrogates the DMV's DNS server with this Bedoop data, and receives in reply the name of the person depicted in the photograph. (If the Bedoop system is a telephone, the name may be displayed on a small LCD display commonly provided on telephones.)

The guard checks the name returned by the Bedoop system with the name printed on the badge. On seeing that the printed and Bedoop-decoded names match (and optionally checking the door log to see that a person of that name was authorized to enter and did so), the security guard can let the candidate pass.

It will be recognized that the just-described arrangement offers very high security, yet this security is achieved without the candidate ever previously visiting the employer, without the employer knowing what the candidate looks like, and by use of an access badge produced by the candidate herself.

Variants of such home-printed badge embodiments find numerous applications. Consider purchasing movie- or event-tickets over the web. The user can print an access ticket that has an entry code embedded therein. On arriving at the theater or event, the user presents the ticket to an optical scanning device, which decodes the entry code, checks the validity of same, authorizes the entry, and marks that entry code as having been used (preventing multiple uses of tickets printed with the same code).

Another Controlled Access ID

A great variety of access control systems can be implemented using the present technology. The foregoing is just one example.

Another application employs an ID card, Bedoop technology, and proximity detection technology (commonly known as RFID).

The ID card can be a badge or the like having a steganographically-encoded photograph of the bearer. The card further includes a proximity ID device, such as an unpowered electronic circuit that is excited and detected by a radiant field from an associated proximity detector, providing a unique signature signal identifying a particular individual.

The building can be provided with an image sensor (such as a video camera or the like), an associated Bedoop detection system, and the proximity detector. When a user wearing the badge approaches, the proximity detector signals the camera to capture image data. The Bedoop detection system identifies the badge photograph (e.g., by clues as are described in the prior applications, or without such aids), captures optical data, and decodes same to extract the steganographically-embedded data hidden therein. The access control system then checks whether the badge ID discerned from the proximity sensor properly corresponds to the Bedoop data extracted from the photograph on the badge. If so, access is granted; if not, the data is logged and an alarm is sounded.

By such arrangement, premises security is increased. No longer can proximity-based access badges be altered to substitute the picture of a different individual. If the photo is swapped, the

proximity system ID and the embedded photo data will not match, flagging an unauthorized attempted access.

The same principles are applicable in many other contexts – not limited to RF-based proximity detection systems. For example, the data decoded from the photograph can be compared against other forms of machine-sensed personal identification associated with the badge. These include, but are not limited to, bar code IDs, mag-stripe ID cards, smart cards, etc. Or the comparison can be with an identification metric not associated with the badge (e.g., retinal scan, voice print, or other biometric data).

Ink-Jet Printing

In the foregoing discussions, reference has been made to use of ink-jet printing as a means for providing steganographically encoded indicia on substrates. The following discussion expands on some of the operative principles.

The basic physics and very low level analog electronic operation of ink-jet printers (sometimes termed bubble-jet printers) are ideally suited to support very-light-tint background digital watermarking on any form of substrate. (Watermarking through apparent "tinting" of substrates is discussed in published specification WO0007356, corresponding to US application 09/127,502.) In general, the statement, "if you can print it with an ink jet printer, you can watermark it" is largely accurate, even for (perhaps especially for) simple text documents. Indeed, there is a degree of flexibility and control in the ink-jet printing realm that is not as generally available in more traditional printing technologies, such as commercial offset printing and other plate-based technologies. (This is not to say that ink-jet has better quality than plate-based technologies; it has more to do with the statistics of ink droplets than anything else.) Heavier tint backgrounds are possible as well, where the continuum ranges from very light background tinting, where the casual observer will see "white paper," all the way through heavily inked patterned backgrounds, and photographs themselves, and everything in between.

In some embodiments, the ink-jet driver software is modified to provide lower-level control of individual droplet emission than is provided in existing printer drivers, which are naturally optimized for text and graphics. In some such embodiments, the "watermarking" print mode is another option from which the user can select (e.g., in addition to High Quality, Econo-Fast, etc.), or the selection can be made automatically by application software that is printing watermarked data.

In more sophisticated embodiments, the watermark data is applied to the printer driver software independently of the other image/text data. The printer driver is arranged to eject droplets in the usual print density for the image/text data, and at a more accurately-controlled, finer density for the separately-applied watermark data. (The latter may be effected as a slight modulation signal on the former.) This arrangement provides for essentially transparent integration into existing printer environments – no one need worry about the watermarking capability except the software applications that specifically make use of same.

Consumer Marking of Web-Based Materials

Various items of printed media can originate off the web, yet be printed at home. Examples include movie tickets, coupons, car brochures, etc. Bedoop data can be added, or modified, by the software application or by the printer driver at the time of printing. (Alternatively, the Bedoop data can be customized to correspond to the user before being downloaded to the user's system for printing.)

One advantage to Bedoop-encoding printed images locally, as opposed to Bedoop-encoding the image files prior to downloading for local printing, is that the encoding can be tailored in accordance with the particular properties of the local printer (e.g., to increase robustness or decrease visibility) – properties not generally known to a remote server.

In one particular example, the UID field in the Bedoop data can be written with a value that serves as an index to a database of user profiles, permitting later systems to which the printed item is presented to personalize their response in accordance with the profile data.

In another example, the UID field serves an authentication purpose, e.g., to verify that the printed medium actually was printed at a particular place, or by a particular user or at a particular time.

Coffee Mug

At retail coffee outlets, customers commonly order the same drink day after day ("half-decaf, short, skinny latte"). Some customers present personal coffee mugs to the cashier, preferring the sensation of ceramic or metal to paper, and avoiding the trash/recycle dilemma.

The drinker's "regular" order can be Bedoop-encoded either on the mug itself or, more commonly, on an adhesive label applied to the mug. The encoding can be in addition to other aesthetic imagery (e.g., artwork or a photo), or the marking can be purely data. Labels the size of postage stamps may be used.

On handing the mug to the cashier, the customer can simply say "the regular." The cashier passes the mug in front of the optical scanning device of a Bedoop system associated with the cash register. The system steganographically decodes the data and provides the corresponding order ("half-decaf, short, skinny latte"), either textually or audibly (e.g., by a voice synthesizer) to the cashier or the barrista. The cash register system also knows the current price of the requested drink, and rings up the charge accordingly.

Labels of the type described can be available to the cashier on pre-printed rolls, just as with other adhesive stickers, or can be printed on-demand. (Small label printers may be best suited in the latter case, given space constraints in retail outlets.) Customers ordering drinks for personal mugs may be invited to take a label corresponding to their just-ordered drink and apply it to their mug for future use.

In variants on this basic theme, the mug label can be further encoded (or a supplemental label can be provided and encoded) with electronic payment information, such as the customer's credit card number, or the number of a debit account maintained by the coffee merchant for that customer. When the mug is scanned for the drink order, the system likewise detects the payment information and charges

the corresponding fee to the appropriate account. (For security reasons, the system may be arranged so that the mug cannot be used to authorize more than, say $5 of coffee drink purchases per day.)

In another variant on this theme, the system maintains an electronic log of coffee purchases made by the customer and, in accordance with then-prevailing marketing considerations, rewards the customer with a free drink after 8 or 12, etc., drinks have been purchased.

In still another variant on this theme, regular customers who use Bedoop-labeled mugs can participate in periodic promotions in which, for example, every N^{th} such customer is rewarded with a cash or merchandise prize. Bells go off when the Nth mug is scanned. (N can be a fixed number, such as 500, or can be a random number – typically within a known range or with a known mean.)

Warping and Focus Issues

The coffee cup is an example of a non-planar object. Another is soft drink cans. Special issues can arise when encoding and decoding markings on such objects. For example, when sensing such an object with a camera or the like, part of the image will be out of focus due to differing distances from the camera to different parts of the can surface.

While parts of an image sensed from a non-planar object, such as a can, may be out of focus, they still convey useful image data. The out of focus areas are just blurred – as if filtered by a low pass filter. But to make use of this information, a further complication must first be addressed: warping.

When viewed from a camera, the planar artwork with which the can is wrapped becomes warped. Portions of the can nearest the camera appear at a nominal full scale, while areas successively further around the can curvature (as viewed from the camera) appear progressively more and more spatially compressed. Regardless of the watermarking technology being employed, the physical warp of the can's surface is likewise manifested as a warping of the encoded watermark data.

On way of handling this issue is to pre-warp the watermark pattern to account for this optical distortion.

In watermarking techniques that operate directly on luminance values, the grid by which the watermark is applied can be pre-distorted to counteract the subsequent optical distortion of the artwork as it is perceived on the cylindrical can. Consider a Pepsi or Coke can. A virtual center line may pass through the center of the logo artwork (the "front" of the can), and serves as a center line of one of the watermark gridded tiles. On either side of this center line the grid is successively stretched. This stretching is computed so that, when viewed by a camera, the watermark grid has the appearance of being uniformly rectilinear, instead of being successively compressed towards the apparent edge of the can, as would otherwise be the case.

An illustration of such an approach is shown in the Figures. Fig. 17 shows an unwarped grid – as would commonly be used in watermarking planar objects. Fig. 18 shows the same grid – prewarped to account for optical distortion due to can curvature. The artwork (e.g., label) center-line is shown by the dotted lines.

More typically the grid is square rather than rectangular. Moreover, the illustrated distortion contemplates that the depicted grid spans +/- 90 degrees from the can front. Also, the grid is typically smaller (e.g., one inch on a side), so several grids are tiled adjacent each other in a span of +/- 90 degrees.) Still further, the illustrated pre-warping is based on an infinite projection (i.e., the can surface as viewed from a distance of infinity – encompassing a full +/- 90 degrees from the center line). More typically, the warp would be computed based on a finite projection – using a typical lens-to-object distance (e.g., 2-24 inches), resulting in a view that encompasses less than a full +/- 90 degree range from the center line.

The illustrated grid is pre-warped only in the horizontal direction, and only in accordance with a curvature-induced geometrical distortion. Another apparent geometrical distortion is also present – one due to different parts of the can being further from the camera. The further away, the smaller the appearance. Accordingly, grid elements expected to be positioned further from the camera should be made commensurately larger in order to pre-compensate for this distance-induced geometrical distortion. Such distance-induced geometrical distortion is manifested equally in the horizontal and vertical directions. Thus, a more accurate pre-warp would also progressively swell the grid cells both in vertical and horizontal dimensions at progressive displacements from the center line, so as to counter-act the further-looks-smaller effect. Fig. 19 shows the basic nature of such a pre-warp.

The degree of this latter pre-warping will be heavily dependent on the distance from the camera lens to the front of the can. If the distance is on the order of two inches, the further-looks-smaller effect will be much more pronounced that if the distance is a foot or more. In the latter case, the distance to the most remote portion of the imaged object may be 110% the distance to the closest portion, whereas in the former case, the ratio may be more on the order of 200%.

The illustrated pre-warping is exemplary of that which may be applied when the watermark is applied in the pixel domain using a grid pattern. It is geometrical pre-warping – i.e., in the spatial domain. Other watermarking approaches would naturally require pre-warping of other sorts, corresponding to the anticipated warping of the watermark data representation. For example, watermarking techniques that rely on changing image coefficients in transformed domains would require different adjustments. Conceptually these adjustments are the same (i.e., resulting in an apparent image having the intended watermark information), but the manifestation is not susceptible to illustration like that given in Figs. 17 and 18 (i.e., since transform coefficients are being changed - rather than grid layout, warping in a transform domain - rather than in a spatial domain, would be required).

Although the foregoing description focused on pre-warping the image, the problem can be handled otherwise. If a rectilinear watermark – not pre-warped – is applied to a cylindrical can, the watermark detector can apply an unwarping operation to counteract the applicable distortion. That is, the detector can virtually remap the raw pixel data to effectively stretch the pixels away from the center line to restore them to their proper rectilinear relationship.

In one embodiment, the sensed image data is first trial-decoded without unwarping – assuming the imaged subject is planar. If no watermark is detected, the same data (or a subsequent frame of image

data) is trial-unwarped to see if perhaps the data so yields a readable watermark. Several successive unwarpings of different characters may be tried. In some embodiments, a detector may continuously cycle through several different unwarping functions (including no unwarping) to try and happen on an unwarping function that permits a watermark be discerned from the image data.

If the application permits, the user may specify the shape of the object so that a single, or limited range, of unwarping functions is applied. Or the user can simply provide a gross cue to the detector (e.g., by selecting between "magazines" or "grocery products' on a user interface associated with the watermark detector). In the former case, the medium is known to be flexible and may assume random simple curvatures other than planar. In such case the detector may spend most of its time trying to decode the watermark assuming the imaged page is planar, and occasionally try applying one of four or eight different unwarping functions as would be appropriate if the magazine page were slightly drooping in different directions. In the latter case, grocery products are generally fairly unflexible and thus have relatively predictable shapes - most commonly planar or cylindrical. In such case the detector may spend half its time trying to decode assuming the object is planar, and spend the other half of its time cycling among a variety of cylindrical unwarping functions.

While the foregoing discussion particularly addressed image watermarking, counterparts of these principles are likewise applicable to audio watermarking.

Smart Elevators

In accordance with another embodiment, a building elevator is provided with one or more optical capture devices. Each device examines monitors the contents of the elevator chamber, looking for Bedoop encoded objects, such as ID badges.

On sensing a Bedoop-encoded object, the elevator can determine – among other data – the floor on which the wearer's office is located. The system can then automatically direct the elevator to that floor, without the need for the person to operate any buttons. (The elevator's button panel can be provided with a new, override button that can be operated to un-select the most recently selected floor(s), e.g., in case a user wants to travel to a different floor.)

To aid in identification, the Bedoop objects (e.g., badges) can be colored a distinctive color, permitting the system to more easily identify candidate objects from other items within the optical capture devices' field of view. Or the object can be provided with a retro-reflective coating, and the elevator can be equipped with one or more illumination sources of known spectral or temporal quality (e.g., constant infra red, or constant illumination with a single- or multi-line spectrum, or a pulsed light source of known periodicity; LEDs or semiconductor lasers, each with an associated diffuser, can be used for each the foregoing and can be paired with the image capture devices). Other such tell-tale clues can likewise be used to aid in object location. In all such cases, the optical capture device can sense the tell-tale clue(s) using a wide field of view sensor. The device can then be physically or electronically steered, and/or zoomed, to acquire a higher resolution image of the digitally-encoded object suitable for decoding.

Magazines

Magazine (and newspaper) pages can be steganographically encoded with Bedoop data to provide another "paper as portal" experience. As with the earlier-described office document case, the encoded data yields an address to a computer location (e.g., a web page) having the same, or related, content.

In one exemplary embodiment, the blank magazine page stock is Bedoop-encoded prior to printing. The watermarking can be performed by high speed ink-jet devices, which splatter a fine pattern of essentially imperceptible ink droplets across each page. Each page can be differently watermarked so that, on decoding, page 21 of a magazine can be distinguished from page 22 of the same magazine (and page 106 of the June 21, 1999, issue can be distinguished from page 106 of the June 28, 1999, issue). If desired, each page can be further segregated into regions – either in accordance with the actual boundaries of articles that will later be printed on the pages, or in a grid pattern, e.g., of 3 columns across by 5 rows high. Each region conveys a distinct Bedoop code, permitting different portions of the page to lead to different web data.)

After watermarking and printing, the pages thus produced are bound in the usual fashion with others to form the finished magazine. (Not all pages in the magazine need to be watermarked.)

Of course, the watermarking can be effected by processes other than ink-jet printing. For example, texturing by pressure rollers is another option well suited for the large volumes of paper to be processed. Or the artwork presented in the advertisement can be digitally watermarked using commercial watermarking software (as is available, e.g., with Adobe Photoshop and Corel image editing products).

On presenting a magazine to the optical scanner device of a Bedoop-compliant computer, the computer senses the Bedoop data, decodes same, and launches a web browser to an internet address corresponding to the Bedoop data. If the magazine page is an advertisement, the internet address can provide information complementary to the advertisement. For example, if the magazine page is an advertisement for a grocery item, the Bedoop data can identify a web page on which recipes that make use of the advertised item are presented. If the magazine page includes a photo of a tropical beach, the Bedoop data can lead to a travel web page (e.g., hosted by Expedia or other travel enterprise) that presents fare and lodging information useful to a reader who wants to vacation at the illustrated beach. (The fare information can be customized to the reader's home airport by reference to user profile data stored on the user's computer and relayed to the web site to permit customization of the displayed page.)

The data to which the Bedoop data leads needn't be static; it can be updated on a weekly, daily, or other basis. Thus, if a months-old magazine page is presented to a Bedoop device, the resultant data can be up-to-the-minute. The linked data can include audio and/or video presentations.

In the case of advertising, the inclusion of Bedoop data increases the value of the ad to the advertiser, and so merits a higher charge to the advertiser from the magazine publisher. This higher

charge may be shared with the enterprise(s) that provides the Bedoop technology and infrastructure through which the higher value is achieved.

Business Card Applications

Conventional business cards can be steganographically encoded with Bedoop data, e.g., by texturing, watermark tinting, ink-jet splattering, text steganography, etc. As with many of the earlier-described embodiments, the steganographic encoding is tailored to facilitate decoding in the presence of arbitrary rotation or scale distortion of the card introduced during scanning. (Some such techniques are shown, e.g., in applicant's earlier patents and publications identified above. Various other techniques are known to artisans.)

When a recipient of a business card holds it in front of a Bedoop sensor, the operating system on the local system launches a local Bedoop application. That local Bedoop application, in turn, establishes an external internet connection to a remote business card server. The address of that server may already be known to the local Bedoop application (e.g., having been stored from previous use), or the local Bedoop system can traverse the above-described public network of DNS servers to reach the business card server.

A database on the business card name server maintains a large collection of business card data, one database record per UID. When that server receives Bedoop data from a local Bedoop system, it parses out the UID and accesses the corresponding database record. This record typically includes more information than is commonly printed on conventional business cards. Sample fields from the record may include, for example, name, title, office phone, office fax, home phone, home fax, cellular phone, email address, company name, corporate web page address, personal web page address, secretary's name, spouse's name, and birthday. This record is transmitted back to the originating Bedoop system.

The local Bedoop system now has the data, but needs further instruction from the user as to how it should be processed. Should a telephone number be dialed? Should the information be entered into a personal contact manager database (e.g., Outlook) on the local system? Etc.

In an exemplary embodiment, the local system presents the available choices to the user, e.g., by textual prompts, synthesized voice, etc. The user responds by manipulating the business card in a manner prompted by the system (e.g., move down to telephone office; move up to telephone at home; move right to access corporate web page; move left to access personal web page; rotate left to enter certain elements from the database record (filtered in accordance with a template) into personal contact manager database, etc.. The local Bedoop system responds accordingly.

Some card givers may choose to make additional information available to card recipients -- information beyond that known in prior art contact-management software applications. For example, one of the choices presented by a local Bedoop system in response to presentation of a business card may be to review the card-giver's personal calendar. (The card-giver can maintain his or her personal calendar on a web-accessible computer.) By such arrangement, the card-recipient can learn when the card-giver may be found in the office, when appointments might be scheduled, etc., etc.

Typically, access to this web-calendar is not available to casual web browsers, but is accessible only in response to Bedoop data (which may thus be regarded as a form of authentication or password data).

Some users may carry several differently-encoded cards, each with a different level of access authorization (e.g., with different UIDs). Thus, some cards may access a biographical page without any calendar information, other cards may access the same or different page with access enabled to today's calendar, or this week's calendar, only, and still other cards (e.g., the "spouse" card) may access the same or different page with access enabled for the card-giver's complete calendar. The user can distribute these different cards to different persons in accordance with the amount of personal information desired to be shared with each.

In accordance with a related embodiment, the database record corresponding to Bedoop business card data can include a "now" telephone number field. This field can be continually-updated throughout the day with the then-most-suitable communications channel to the card-giver. When the card-giver leaves home to go to the office, or leaves the office for a trip in the car, or works a week at a corporate office in another town, etc., this data field can be updated accordingly. (A pocket GPS receiver, with a wireless uplink, can be carried by the person to aid in switching the "now" number among various known possibilities depending on the person's instantaneous position.) When this database record is polled for the "now" number, it provides the then-current information.

Consider a Bedoop-enabled public telephone. To dial the phone, a business card is held in front of the Bedoop sensor (or slid through an optical scanner track). The phone interrogates the database at the business card server for the "now" number and dials that number.

To update the any of the fields stored in the database record, the card giver can use a special card that conveys write-authorization privileges. This special card can be a specially encoded version of the business card, or can be another object unique to the card-giver (e.g., the card-giver's driver's license).

The reference to business cards and personal calendars is illustrative only. Going back a century, "calling cards" were used by persons whose interests were strictly social, rather than business. The just-discussed principles can be similarly applied. Teenagers can carry small cards to exchange with new acquaintances to grant access to private dossiers of personal information, favorite music, artwork, video clips, etc. The cards can be decorated with art or other indicia that can serve purposes wholly unrelated to the Bedoop data steganographically encoded therein.

Gestural Input

A Bedoop system can determine the scale state, rotation state, X-Y offset, and differential scale state, of an object by reference to embedded calibration data, or other techniques. If the scan device operates at a suitably high frame rate (e.g., five or ten frames per second), change(s) in any or all of these four variables can be tracked over time, and can serve as additional input.

In an earlier-discussed example, moving an object to the left or right in front of the Bedoop scanner caused a left- or right-positioned button in a dialog box to be selected. This is a change in the X-Y offset of the scanned object. In that earlier example, moving the object inwardly towards the camera caused the selected button to be activated. This is a change in the scale state of the scanned object.

In similar fashion, twisting the object to the left or right can prompt one of two further responses in a suitably programmed Bedoop application. (This is a change in the rotation state.) Likewise, tilting the object so that one part is moved towards or away from the camera can prompt one of two further responses in the application. (This is a change in the differential scale state.)

In the business card case just-discussed, for example, the card can be held in front of the Bedoop scanner of a computer. If the card is twisted to the left, the computer opens a web browser to a web page address corresponding to Bedoop data on the card. If the card is twisted to the right, the computer opens an e-mail template, pre-addressed to an e-mail address indicated by the card.

In other examples, twisting an object to move the right edge towards the scanner can be used to effect a right mouse click input, and twisting the object to move the right edge away from the scanner can be used to effect a left mouse click input.

Simultaneous changes in two of these four positioning variables can be used to provide one of four different inputs to the computer (e.g., (a) twisting left while moving in; (b) twisting left while moving out; (c) twisting right while moving in; and (d) twisting right while moving out). Simultaneous changes to three or all four of these variables can similarly be used to provide one of eight or sixteen different inputs to the computer.

Simultaneous manipulations of the object in two or more of these modes is generally unwieldy, and loses the simple, intuitive, feel that characterizes manipulation of the object in one mode. However, a similar effect can be achieved by sequential, rather than simultaneous, manipulation of the card in different modes (e.g., twist left, then move in). Moreover, sequential manipulations permit the same mode to be used twice in succession (e.g., move in, then move out). By such sequential manipulations of the object, arbitrarily complex input can be conveyed to the Bedoop system.

(It will be recognized that a digitally-encoded object is not necessary to the gestural-input applications described above. Any object (talisman) that can be distinguished in the image data can be manipulated by a user in the manners described above, and an appropriate system can recognize the movement of the object and respond accordingly. The provision of digital data on the object provides a further dimension of functionality (e.g., permitting the same gesture to mean different things, depending on the digital encoding of the object being manipulated), but this is not essential.

Moreover, even within the realm of digitally-encoded gestural talismans, steganographic encoding is not essential. Any other known form of optically-recognizable digital encoding (e.g., 1D and 2D bar codes, etc.) can readily be employed.

In an illustrative embodiment, a business card or photograph is used as the talisman, but the range of possible talismans is essentially unlimited.

Dynamic gestures are not the only communications that can be effected by such talismans. Static positioning (e.g., presenting the talisman at different orientations) can alternatively be employed.

Consider a magazine advertisement. When presented to the sensor with the top of the page up, a first response can be invoked. If the page is presented at a rotation of 90 degrees, a second response can be invoked. Similarly with 180 degrees rotation (i.e., upside down), and 270 degrees rotation. The Bedoop detector can detect these different rotational states by reference to attributes of the watermark signal discerned from the magazine page (e.g., by reference to the rotation state discerned from the subliminal grid signal detailed in applicant's prior patents).

Gestural Decoding Module

There are various ways in which the Bedoop system's decoding of gestural input can be effected. In some Bedoop systems, this functionality is provided as part of the Bedoop applications. Generally, however, the applications must be provided with the raw frame data in order to discern the gestural movements. Since this functionality is typically utilized by many Bedoop applications, it is generally preferable to provide a single set of gestural interpretation software functions (commonly at the operating system level) to analyze the frame data, and make available gestural output data in standardized form to all Bedoop applications.

In one such system, a gestural decoding module tracks the encoded object within the series of image data frames, and outputs various parameters characterizing the object's position and manipulation over time. Two of these parameters indicate the X-Y position of the object within current frame of image data. The module can identify a reference point (or several) on the object, and output two corresponding position data (X and Y). The first represents the horizontal offset of the reference point from the center of the image frame, represented as a percentage of frame width. A two's complement representation, or other representation capable of expressing both positive and negative values, can be used so that this parameter has a positive value if the reference point is right of center-frame, and has a negative value if the reference point is left of center frame. The second parameter, Y, similarly characterizes the position of the reference point above or below center-frame (with above-being represented by a positive value). Each of these two parameters can be expressed as a seven-bit byte. A new pair of X, Y parameters is output from the gestural decoding module each time a new frame of image data is processed.

In many applications, the absolute X-Y position of the object is not important. Rather, it is the movement of the object in X and Y from frame-to-frame that controls some aspect of the system's response. The Bedoop application can monitor the change in the two above-described parameters, frame to frame, to discern such movement. More commonly, however, the gestural decoding module performs this function and outputs two further parameters, X' and Y'. The former indicates the movement of the reference point in right/left directions since the last image frame, as a percentage of the full-frame width. Again, this parameter is represented in two's complement form, with positive values representing movement in the rightward direction, and negative values representing movement in the leftward

direction. The later parameter similarly indicates the movement of the reference point in up/down directions since the last frame.

The scale, differential scale, and rotation states of the object can be similarly analyzed and represented by parameters output from the gestural decoding module.

Scale state can be discerned by reference to two (or more) reference points on the object (e.g., diagonal corners of a card). The distance between the two points (or the area circumscribed by three or more points) is discerned, and expressed as a percentage of the diagonal size of the image frame (or its area). A single output parameter, A, which may be a seven-bit binary representation, is output.

As with X-Y data, the gestural decoding module can likewise monitor changes in the scale state parameter since the last frame, and product a corresponding output parameter A'. This parameter can be expressed in two's complement form, with positive values indicating movement of the object towards the sensor since the last frame, and negative values indicating movement away.

A differential scale parameter, B, can be discerned by reference to four reference points on the object (e.g., center points on the four edges of a card). The two points on the side edges of the card define a horizontal line; the two points on the top and bottom edges of the card define a vertical line. The ratio of the two line lengths is a measure of differential scale. This ratio can be expressed as the shorter line's length as a percentage of the longer line's length (i.e., the ratio is always between zero and one). Again, a two's complement seven-bit representation can be used, with positive values indicating that the vertical line is shorter, and negative values indicating that the horizontal line is shorter. (As before, a dynamic parameter B' can also be discerned to express the change in the differential scale parameter B since the last frame, again in two's complement, seven bit form.)

A rotation state parameter C can be discerned by the angular orientation of a line defined by two reference points on the object (e.g., center points on the two side edges of a card). This parameter can be encoded as a seven-bit binary value representing the percentage of rotational offset in a clockwise direction from a reference orientation (e.g., horizontal). (The two reference points must be distinguishable from each other regardless of angular position of the object, if data in the full range of 0 – 360 degrees is to be represented. If these two points are not distinguishable, it may only be possible to represent data in the range of 0-180 degrees.) As before, a dynamic parameter C' can also be discerned to express the change in the rotation state parameter C since the last frame. This parameter can be in seven bit, two's complement form, with positive values indicating change in a clockwise rotation

The foregoing analysis techniques, and representation metrics, are of course illustrative only. The artisan will recognize many other arrangements that can meet the needs of the particular Bedoop applications being served.

In the illustrative system, the Bedoop application programs communicate with the gestural decoding module through a standardized set of interface protocols, such as APIs. One API can query the gestural input module for some or all of the current position parameters (e.g., any or all of X, Y, A, B, and C). The module responds to the calling application with the requested parameter(s). Another API can query the gestural input module for some or all of the current movement data (e.g., any or all of X',

Y', A', B' and C'). Still another API can request the gestural decoding module to provide updated values for some or all of the position or movement data on a running basis, as soon as they are discerned from each frame. A complementary API discontinues the foregoing operation. By such arrangement, all of the gestural data is available, but the Bedoop application programs only obtain the particular data they need, and only when they ask for it.

In Bedoop applications that communicate with external servers, just the Bedoop data (i.e., CLASS, DNS, and optionally UID) may initially be sent. If the remote server needs to consider gestural data in deciding how to respond, the remote server can poll the local Bedoop system for the necessary data. The requested gestural data is then sent by the local Bedoop system to the remote server in one or more separate transmissions.

In other embodiments, since the gestural data is of such low bandwidth (e.g., roughly 56 bits per image frame), it may routinely and automatically be sent to the remote computer, so that the gesture data is immediately available in case it is needed. In an illustrative implementation, this data is assembled into an 8-byte packet, with the first byte of the packet (e.g., the X parameter) being prefixed with a "1" sync bit, and subsequent bytes of the packet being prefixed with "0" sync bits. (The sync bits can be used to aid in accurate packet decoding.)

In some embodiments, it is useful to provide for an extension to the normal 64-bit Bedoop length to accommodate an associated packet of gestural data. This can be effected by use of a reserved bit, e.g., in the UID field of the Bedoop packet. This bit normally has a "0" value. If it has a "1" value, that indicates that the Bedoop data isn't just the usual 64 bits, but instead is 128 bits, with the latter 64 bits comprising a packet of gestural data.

Similar extension protocols can be used to associate other ancillary data with Bedoop data. A different reserved bit in the UID field, for example, may signal that a further data field of 256 bits follows the Bedoop data – a data field that will be interpreted by the remote computer that ultimately services the Bedoop data in a known manner. (Such bits may convey, e.g., profile data, credit card data, etc.) The appended data field, in turn, may include one or more bits signaling the presence of still further appended data.

Grandmothers

It is a common complaint that computers are too complex for most people. Attempts to simplify computer-user interaction to facilitate use by less experienced users usually serve to frustrate more experienced users.

In accordance with this aspect of the present technology, the sophistication of a computer user is steganographically indicated on a talisman used by that user to interact with the system. The computer detects this steganographically-encoded data, and alters its mode of interacting with the user accordingly.

Consider internet browser software. Experienced users are familiar with the different functionality that can be accessed, e.g., by various drop-down menus/sub-menus, by the keyboard

shortcuts, by the menus available via right-clicking on the mouse, by manipulating the roller mouse scroll wheel and scroll button, etc., etc. Grandmothers of such users, typically, are not so familiar.

Although gestural interfaces hold great promise for simplifying user-computer interaction, the same dichotomy between experienced users and inexperienced users is likely to persist, frustrating one class of user or the other.

To help close this gap, a computer system can respond to gestures in different manners, depending on the expertise level indicated by encoding of the talisman. For an expert user, for example, the gestural interface active in the internet browser software may display the stored list of Favorite web addresses in response to tipping the left edge of the talisman towards the optical sensor. Once this list is displayed, the expert user may rotate the talisman to the right to cause the highlighting to scroll down the list from the top. Rotating the talisman to the left may scroll the list of Favorites up from the bottom. The speed of scrolling can be varied in accordance with the degree of rotation of the talisman from a default orientation.

In contrast, for the novice user, these talisman manipulations may be confounding rather than empowering. Tipping the left edge of the talisman towards the sensor may occur as often by mistake as on purpose. For such users, a more satisfactory interface may be provided by relying on simple X-Y movement of the talisman to move an on-screen cursor, with a movement of the talisman towards the sensor to serve as a selection signal (i.e., like a left-mouse click).

(In the example just-cited, the expert user summoned a list of Favorite web sites. Different "Favorites" lists can be maintained by the computer – each in association with different talismans. A husband who uses one talisman is provided a different "Favorites" list than a wife who uses a different talisman.)

Printed Pictures

A printed photograph can be steganographically encoded with Bedoop data leading to information relating to the depicted person (e.g., contact information, biographical information, etc.).

Such a photograph can be presented to a Bedoop sensor on a telephone. In a simple embodiment, the telephone simply processes the Bedoop data to obtain a corresponding default telephone number, and dials the number. In other embodiments, various options are possible, e.g., dial home number or dial work number. On presenting the photograph to the telephone, for example, moving the photo to the left may dial the person at home, while moving the photo to the right may dial the person at work.

As telephones evolve into more capable, multi-function devices, other manipulations can invoke other actions. In a computer/telephone hybrid device, for example, rotating the photo counterclockwise may launch a web browser to an address at which video data from a web cam at the pictured person's home is presented. Rotating the photo clockwise may present an e-mail form, pre-addressed to the e-mail address of the depicted person. Moving the photo to the right may query a database on the system

for other photographs depicting the same individual or subject, which can be presented in response to further user input. Etc.

In this and other embodiments, it is helpful for the Bedoop device to prompt the user to aid in manipulating the object. This can be done audibly (e.g., "move photo left to dial at home") or by visual clues (e.g., presenting left- or right-pointing arrows).

Bedoop data in photographs can also be used to annotate the photographs, as with notes on the back of a photograph, or printed under the photograph in a photo album. The Bedoop data can lead to a remote database, where the photograph owner is permitted to enter a textual (or audio) narrative in association with each photograph's UID. Years later, when some of the names have been forgotten, the photograph can be positioned in front of a Bedoop sensor, and the system responds by providing the annotation provided by the photograph owner years earlier.

Drivers Licenses and Other Cards

Drivers licenses, social security cards, or other identity documents may be encoded by the issuing authority with Bedoop data that permits access to the holder's personal records over the web. On presenting the document to a Bedoop system, the system directs a web browser to a private address corresponding to data encoded on the document. At that address, the holder of the document can review governmental records, such as state or federal tax return data, social security entitlements, etc., as well as privately-maintained records, such as credit records, etc. User selection among various functions can be effected by spatial manipulation of the document. (Entry of additional data, such as social security number or mother's maiden name, may be required of the user to assure privacy in case the document is lost or stolen.)

By manipulating a driver's license in front of a Bedoop sensor, a user can request renewal of the driver's license, and authorize payment of the corresponding fee.

Bank cards (debit, credit, etc.) can similarly be encoded with Bedoop data to permit the holder to access bank records corresponding to the bank card account. (Entry of a PIN code may be required to assure privacy.)

Such documents can also be used to access other personal data. One example is e-mail. A traveler might pause at a Bedoop kiosk at an airport and present a driver's license. Without anything more, the kiosk may present email that is waiting for the traveler on an associated display screen.

On recognizing a driver's license, the kiosk can access a remote site (which may be maintained by the Department of Motor vehicles, another government entity, a private entity, or by the traveler), authenticating the operation by presenting Bedoop data encoded on the license, and obtain information that the person has pre-approved for release in response to such authorized access. This information can include e-mail account and password information. Using this information, the kiosk queries the corresponding e-mail server, and downloads a copy of recently received mail for presentation at the kiosk. (A user-entered PIN number may be required at some point in the process, e.g., in querying the

remote site for sensitive e-mail password data, before presenting the downloaded e-mail for viewing, etc., to ensure privacy.)

Other cards carried in wallets and purses can also be encoded to enable various functions. The local sandwich shop that rewards regular customers by awarding a free sandwich after a dozen have been purchased can encode their frequent-buyer card with Bedoop data leading to the shop's web-based sandwich delivery service. Or the frequent-buyer card can be eliminated, and customers can instead wave their business card or other identity document in front of the shop's Bedoop sensor to get purchase credit in a tally maintained by the sandwich shop's computer.

Food stamps, health insurance cards, and written medical prescriptions, can likewise be encoded with digital data to enable the provision of new functionality.

At large trade shows, such as COMDEX, vendors needn't publish thick, glossy brochures to hand out to visitors. Instead, they may print various stylish promo cards for distribution. When later presented to a Bedoop sensor, each card leads to a web-based presentation – optionally including persuasive video and other multi-media components. The user can be prompted to provide data to customize, or focus, the presentation to the user's particular requirements. If the user wants further information, a request can be made by the click of a mouse (or the twist of a card).

Prizes and Product Promotions

Product packaging (e.g., Coke cans, Snapple bottles, Pepsi 12-pack boxes) can be encoded for contest purposes. The encoding can be customized, item to item, so that selected items – when Bedoop scanned – are recognized to be the one in a hundred that entitles the owner to a cash or merchandise prize. A remote server to which the item's Bedoop data is provided queries the user for contact information (e.g., address, phone number) so the prize can be awarded or, for smaller prizes, the system can print out an award certificate redeemable at local merchants for products or cash. Once a winning item is identified to the remote server, its UID on the server is marked as redeemed so that the item cannot later be presented to win another prize.

In other such embodiments, all of the items are encoded identically. Winners are determined randomly. For example, during a contest period, persons around the world may present Coke cans to Bedoop systems. The corresponding Bedoop application on each user computer submits Bedoop data to a corresponding web address. The user's e-mail address may also be included with the submission. As this data is relayed to the corresponding server computer(s), every N^{th} set of data is deemed to be a winner, and a corresponding award notification or prize is dispatched to the Bedoop system from which the winning set of data originated.

The server computer that receives such contest submittals from client Bedoop systems can be arranged to prevent a single user from bombarding the server with multiple sets of data in an attempt to win by brute force. (This may be done, for example, by checking the included e-mail address or received IP address, and not considering a data submittal if the same address was encountered in data submitted within the past hour. Similar anti-brute-force protection can be provided on the user's computer,

preventing, e.g., repeated contest data to be sent more frequently than once per hour. More sophisticated anti-brute-force measures can of course be provided.)

Non-planar product packaging, such as cylindrical soda cans, present certain optical issues in encoding and decoding which are detailed further below.

Product Information and Ordering

Product packaging and product advertisements can be encoded with Bedoop data that, when presented to a Bedoop system, initiates a link to a web page from which that product can be purchased, or more information obtained. Once the link has been established, the user can be instructed to manipulate the object in different of the earlier-described modes to effect different functions, e.g., move towards camera to order the product; move away from camera for product information. If the object is moved towards the camera to effect an order, the user can be prompted to further manipulate the object to specify delivery options (e.g., rotate left for overnight mail, rotate right for regular mail). If the object is moved away from the camera to request product information, the user can be promoted to further manipulate the object to specify the type of information desired (e.g., rotate left for recipes, rotate right for FDA nutritional information, move up for information on other products in this family, move down to send an email to the product manufacturer).

Credit card or other customer billing information, together with mailing address information, can be stored in a profile on the Bedoop system, and relayed to the transactional web site either automatically when a purchase action is invoked, or after the user affirms that such information should be sent (which affirmation may be signaled by manipulation of the packaging or advertisement in one of the earlier-described modes). Other modes of payment can naturally be employed.

Computer Access Cards

This disclosure earlier considered access cards used to gain access to secure buildings. Related principles can be used in conjunction with computer access.

A driver's license, employee photo ID, or other such document can be presented to a Bedoop sensor on a computer. The computer recognizes the user and can take various steps in response.

One response is to log onto a network. Another is to set load a user profile file by which the computer knows how to arrange the desktop in the user's preferred manner. By manipulating the Bedoop-encoded object, the user can further vary the environment (e.g., rotate left to launch standard business productivity applications and software development applications; rotate left to launch lunchtime diversions – stock update, recreational games, etc.)

Hotel rooms are increasingly providing computer services. By presenting a driver's license, a Bedoop-equipped computer in a hotel room can link to a remote site indicated by the Bedoop data, obtain preference data for that user, and launch applications on the hotel computer in an arrangement that mimics that user's familiar work computer environment.

Audio/Video Disks, Software, and Books

Bedoop data can be conveyed by indicia or texturing on the surfaces of CD and DVD disks, on the labels (or authenticity certificates) or inserts or artwork for same, on the enclosures for same (e.g., jewel box, plastic case, etc.), on book dust jackets, on book pages, etc. Any of these objects can be presented to a Bedoop device to establish a link to a related web site. The consumer can then manipulate the object (or otherwise choose) to select different options.

For music, one option is to receive MP3 or other clips of songs by the same artist on other CDs, or of songs from other artists of the same genre. Another is to view music video clips featuring the same artist. Still another is to order tickets to upcoming concerts by that artist. In-store kiosks can permit tentative customers to listen to sample tracks before they buy.

Similar options can be presented for video DVDs. In the case of video, this can include listings of other movies with the same director, with the same star(s), etc. In the case of software, the options can include advisories, bug fixes, product updates and upgrades, etc. Naturally, the user can make purchases from these sites, e.g., of other music by the same artist, other videos with the same star, software upgrades, etc.

Similar options can be accessed using Bedoop data associated with printed book materials.

Children learn the mechanics of turning book pages at an early age. Children learn to look at pictures on the pages of a book and they enjoy hearing the story that is related to the images. Generally, adults read the words and the child follows along looking at the pictures. Children enjoy repeatedly hearing the words of a story. The association of seeing pictures and repeatedly hearing the words is an excellent mechanism for learning to read and learning to enjoy books.

Embedded digital watermark data can automate the above process so that a child can see the pictures, and hear the words independently. Such an arrangement provides enjoyment for the child while teaching reading and love for books, and at the same time giving the child independence and familiarity with an automated mechanism.

More particularly, images in the book, or the paper substrates of the pages, contain digital watermarks. As a child turns the pages of the book, a camera captures the image and an associated computer reads the watermark. The watermark is then used to index a data store – either local or remote – to obtain text corresponding to the page being viewed. A text-to-speech converter is then employed to voice the text to the child. (Alternatively, the data store can contain digitized speech, rather than simple text – permitting different story characters to be given different voices, etc.) Thus, as a child turns the pages of the book, the child hears the words that are printed on the page. By turning pages, the child controls the process. The child will naturally make an association between the printed words and the words that are played by the computer.

Ad Tracking

Advertisers commonly use different advertisements for the same product or service, and employ means to track which ad is more effective within which demographic group. Bedoop can provide such functionality.

Consider a travel service web site that is promoting Hawaiian vacations. Bedoop data from several advertisements can lead consumers to the site.

Identical advertisements can be placed in several different magazines. Each is encoded with a different Bedoop UID. By monitoring the UIDs of the Bedoop inquiries to the site, the travel service can determine which magazines yield the highest consumer response (e.g., per thousand readers).

Likewise, within a single magazine, two or more advertisements may be encoded with Bedoop data leading to the site – again, each with a different UID. Again, analysis of the UIDs used in accessing the site can indicate which advertisement was the more effective.

The two UIDs in the foregoing examples may both lead to the same internet destination, or may lead to different destinations.

The instantaneous nature of the internet links permits advertisers to learn how consumer responses to print advertisements vary with time-of-day, yielding information that may assist in making ads for certain products more effective.

More elaborate variants and combinations of the foregoing are, of course, possible, If the consumers provide personal information in response to the ads (either by permitting access to pre-stored personal profile data, or by filling in web-based forms, or by manipulation of the ad (e.g., "please move the ad towards your Bedoop sensor if you drank coffee this morning")), still richer statistical data can be gleaned.

Rolodex of Cards

Bedoop-encoded business cards as detailed above can be accumulated and kept near the telephone or computer in a Rolodex-like arrangement. If a refrigerator ice-maker malfunctions, a homeowner can find the card for the appliance repairman used a few years ago, and present it to a Bedoop sensor. A link is established to the repairman's company (e.g., web site or via telephone). At a web site, the repairman may provide basic information, such as hours of availability, current fee schedule, etc. The homeowner may select an option (by card gesture or otherwise) to invoke a teleconference (e.g., NetMeeting) to consult about the problem. Or the homeowner may select another option to send e-mail. Still a further option may permit the homeowner to schedule a house call on the repairman's weekly calendar. Still a further option may permit the homeowner to view one or more short videos instructing customers how to fix certain common appliance problems.

Stored Value Cards

An electronic money system (e.g., of the sort detailed in US application 60/134,782, filed May 19, 1999) may encode Bedoop data on a card that leads to storage at which random-number tokens

(which represent increments of money) are stored. Presenting the card to a Bedoop system launches an application that reads and encrypts the tokens and forwards the encrypted data to the clearinghouse computer of the corresponding bank to learn their remaining value. There the tokens are decrypted and checked for validity (but not redeemed). The bank computer responds to the Bedoop system, indicating the remaining value of the tokens on the card.

For security reasons, the storage containing the random-number tokens should not be generally accessible. Instead, the user must provide authentication data indicating authorization to gain access to that information. This authentication data may be a PIN code. Or the user may provide authentication by presenting a second Bedoop-encoded object, e.g., a driver's license to the Bedoop system. (Many other Bedoop systems may advantageously use, or require the use of, two or more Bedoop objects – either presented one after the other, or all at the same time. The Bedoop system can provide visual or audible prompts leading the user to present the further Bedoop object(s) as necessary.

Ski Lift Tickets

In accordance with another embodiment, ski lift tickets are Bedoop encoded to provide various functionality.

For example, instead of buying a lift ticket good for a day, a skier may purchase a ticket good for eight lifts. This data is encoded on the ticket, and sensed by a Bedoop sensor at each lift. The sensors are networked to a common server that tracks the number of lifts actually purchased, and updates the number as used. The skier is informed of the number of rides remaining on entering or leaving the lift. Statistical data can be collected about trail usage (e.g., N% percent of skiers ski all day along just two lifts, etc.).

Off the slopes, back at home, the used lift ticket may be presented to a Bedoop sensor to obtain current snow conditions and lift hours, or to review trail maps, or to order ski vacation packages. If the ticket is encoded with the owner's name, UID, or other information of commercial/marketing interest, local merchants may give the bearer discounts on selected goods in response to Bedoop scanning of the ticket and recovery of such information.

REI Membership Cards

Membership cards for certain stores can be Bedoop-encoded to provide added value to the member. For outdoor gear stores such as REI, presentation of the card to a Bedoop sensor can lead to a library of USGS maps, to web pages with current fishing and hunting regulations, etc. Naturally, the store's on-line ordering site is just a quick twist away.

Theme Park Tickets

Theme park tickets can be encoded with the age and gender of the visitor, and with additional data permitting the experience to be customized (e.g., from a roster of theme park personalities, the visitor's favorite is Indiana Jones). Throughout the park are kiosks to which the visitor can present the

ticket to orchestrate the visit to follow a particular story line. Some kiosks issue premiums matching the age/gender of the recipient.

Car Keys

Car keys (or key ring fobs) are Bedoop encoded. When the car is taken to a shop for service, the mechanic presents the key to a Bedoop sensor, and thereby obtains the car's maintenance history from a remote server on which it is maintained. At home, the key can be presented to a Bedoop sensor and manipulated to navigate through a variety of automotive-related web sites.

In some embodiments, the Bedoop-encoded object is not used to navigate to a site, but is instead used to provide data once a user's computer is otherwise linked to a web site. A user surfing the web who ends up at a car valuation site can present a key to the Bedoop scanner. The Bedoop data is used to access a remote database where the make, model, options, etc., of the car are stored. This data is provided to a database engine that returns to the user the estimated value of the car.

While visiting a mechanic's web site, presentation (and optionally manipulation) of a key or key ring fob can be employed to schedule a service appointment for the car.

Fashion Coordination

Some department stores and clothing retailers offer "personal shoppers" to perform various services. For example, a customer who is purchasing a dress may ask a personal shopper for assistance in selecting shoes or accessories that complement the dress.

A Bedoop-encoded garment tag on the dress can be employed to obtain similar assistance. In response to such a tag, a Bedoop system can query a database to obtain a mini-catalog of clothes and accessories that have previously been identified as complementing the dress identified by the tag. These items can be individually displayed on a screen associated with the system, or a virtual model wearing the dress - together with one or more of the recommended accessories - can be synthesized and depicted. The shopper may quickly review the look achieved by the model wearing the dress with various different pairs of shoes, etc., by repeatedly activating a user interface control (by mouse, touch screen, or garment tag gestures) to cycle through different combinations.

A shopper's credit card can be Bedoop-encoded so as to lead Bedoop systems of particular stores (i.e., stores pre-authorized by the shopper) to a profile on the shopper (e.g., containing size information, repeat purchase information, return history, style/color preferences, etc.).

Credit Card Purchases

When a consumer visits a commercial web site and wishes to purchase a displayed product, the transaction can be speeded simply by presenting a Bedoop-encoded credit card to a Bedoop sensor on the user's computer. The Bedoop data on the card leads to a database entry containing the credit card number and expiration date. The Bedoop application then sends this information (optionally after encrypting same) to the web site with instructions to purchase the depicted product.

(Impulse purchases are commonly deterred by the hurdles posed between the purchase impulse and the completed purchase. This and other Bedoop applications aid in reducing such hurdles.)

Product Marketing

Bedoop data relating to one product or service can be used to cross-market others products and services. Consider a consumer who purchases a pair of golf shoes. The box is Bedoop encoded. By presenting the box to a Bedoop system, the consumer is linked to a web page that presents various promotional offers. The consumer may, for example, elect to play a free round of golf at one or more identified local golf courses, or print a coupon for ten percent off any order of socks from an on-line sock merchant. (Various means can be employed to prevent multiple redemptions from a single box. One is a serial number that is tracked by the web page or cross-marketed merchant, and only honored once. Another is identification data corresponding to the consumer that is tracked to prevent multiple redemptions.)

Product tags can likewise be Bedoop-encoded. A tag from an article of Nike apparel can lead to the Nike on-line store, where the user can buy more merchandise. If the tag is from a soccer jersey, a certain tag manipulation (e.g., rotate left) may lead the user to a special-interest soccer page, such as for the World Cup. A tag on a golf glove may lead to a website of a local golf course. Twist left to reserve a tee time; twist right to review course maps and statistics. Bedoop kiosks can be provided in retail stores to let consumers use the Bedoop features.

Travel Planning Services

After making a reservation at a resort, a consumer is typically mailed (by email or conventional mail) various confirmation information. If not already printed, the consumer can print this information (e.g., a confirmation card).

Bedoop-encoding on the printed object can lead to web-based information relating to the reservation (e.g., reservation number, the consumer's name, arrival/departure dates, etc.). If the consumer wishes to make dinner or golf reservations, this object is presented to a Bedoop system – either at the user's home, at an airport kiosk, etc. The system recognizes the object type and encoded data, and establishes a link to a remote computer that provides various information and scheduling services for the resort. By manipulating the object (or otherwise) the consumer selects desired dinner and golf tee times. The system already has the reservation number (indexed by the UID), so tedious provision of such data is avoided.

In some embodiments, the remote computer is not maintained by the resort, but is rather maintained by an independent travel service. (The travel service may also maintain the DNS leaf node server.) The computer can present a web page (branded by the travel service or not) that offers the scheduling options desired by the user, and also presents links to other information and services (e.g., offering entry tickets to nearby attractions, and advertising nearby restaurants).

Airline tickets (or e-ticket confirmations) can be similarly encoded with Bedoop data. These items may be presented to Bedoop systems – at a traveler's home or in airports – to permit review and changing of travel itinerary, reserve hotels and rental cars, secure first-class upgrades, check the airplane's seating arrangement, review frequent flier status, scan tourist information for the destination, etc.

Movie Tickets

As indicated earlier, movie tickets can be encoded with Bedoop data identifying, e.g., the movie title and date. When a movie viewer returns home, the ticket stub can be presented to a Bedoop system. One of the options presented by the corresponding Bedoop application can be to launch a pay-per-view screening of the just-seen movie at a discounted rate. Another is to download the movie onto a writeable DVD disk at the viewer's home, perhaps serialized to permit playback only on that viewer's DVD player, or enabled for only a few playbacks, etc. (again, likely for a discounted fee). Still another option is to present web-delivered video clips from the movie. Another is to offer related merchandise for purchase, possibly at discount to retail. (These features may be available for only a limited period after the date encoded on the ticket stub.) Another is to alert the consumer to upcoming movies of the same genres, or with the same director or stars, or released by the same studio. Still another is to direct a web browser to an on-line ticket merchant for tickets to other movies. The consumer may navigate among these options by manipulating the ticket stub, or otherwise.

The same, or related, options can likewise be provided in response to Bedoop data detected from a book jacket presented to a Bedoop system.

Video Recording

A video recording device can be programmed to record a broadcast program by presenting a Bedoop sensor with a printed promotion for the program (e.g., an advertisement in a newspaper or TV Guide). Bedoop-encoded within the printed document is data by which the Bedoop system (which may be built into the video recorder or separate) can set the recording time, date, and channel.

Set Top Boxes

Many entertainment-related applications of Bedoop data can be implemented using television set top boxes. Such boxes include processors, and typically include a return channel to a control facility. The provision of a Bedoop chip and optical sensor can vastly increase the functionality these devices presently provide.

Special Event Tickets

Consider a ticket to a basketball game. By presenting the ticket to a Bedoop system, a user may access the web site of either team so as to review recent scores and statistics. The user may also obtain a web-based virtual tour of the arena, and review seating maps. The view of the playing field as seen from

the user's seat location may be presented. Tickets for upcoming games may be ordered, as well as pay-per-view games and team souvenirs. For high-priced tickets, the user may be entitled to premium web features, such as on-line text-, audio-, or video-chat session with a team star on the day before the game.

Unlike conventional tickets, Bedoop-encoded tickets need not limit the user to a predetermined seat. While the ticket may be printed with a nominal seat, the user may present the ticket to a Bedoop sensor and access a web site at which a different seat can be reserved. On attending the event, the consumer presents the ticket to a Bedoop sensor that reads the ticket UID and looks up the seat assignment most-recently picked by the consumer. It then prints a chit entitling the consumer to take the seat earlier selected from the transactional web site.

Signet Rings

Signet rings have historically been used to indicate a person's identity or office. Such rings, or other items of personal jewelry, can be encoded with Bedoop data (either by texturing or printing) and presented as necessary to Bedoop systems. The extracted Bedoop data can lead to a secure web site indicating the person's name and other information (i.e., a web site that has anti-hacking measures to prevent illicit change of the stored identification information). Such a signet ring can be presented to Bedoop systems that require a high-confidence confirmation of identity/authorization before proceeding with a Bedoop function.

Tattoos

Temporary tattoos are well known and typically include an ink or printed film that is transferred to a wearer's skin through some application process (e.g., wetting or otherwise). The artwork on the tattoo can be arranged to steganographically encode Bedoop data, facilitating machine recognition of the persons (or objects) tattooed. Youths can compile a contacts database simply by snapping digital photos of friends using an imaging-capable personal digital assistant. Such a computer device can decode the watermark, access a corresponding web dossier of information on the person photographed, and add such information to a contacts database.

E-Paper

While it is doubtless evident that a great variety of printing technologies can be employed in Bedoop applications, it should be noted that e-paper can be employed as well. E-paper, developed by Nicholas K. Sheridon of Xerox, and mass produced by 3M, is a thin layer of transparent plastic in which millions of small beads, somewhat like toner particles, are dispersed. The beads, each contained in a fluid -filled cavity, are free to rotate within those cavities. The beads are "bichromal," with hemispheres of contrasting color (e.g. black and white). Under the influence of a voltage applied to the surface of the sheet, the beads rotate to present one colored side or the other to the viewer. A pattern of voltages can be applied to the surface in a bit-wise fashion to create images such as text and pictures. The image persists

until new voltage patterns are applied to create new images. The reader is presumed familiar with the US patents issued to Sheridon on this technology.

It will further be recognized that epaper can be used to convey digital data according to essentially any known watermarking method, and is also suitable for conveying digital information in data glyph form.

Post-It® Notes

Pads of Post-It® notes, or other pads of paper, can be marked by the manufacturer (either by texturing, watermarked tinting, ink-jet spattering, etc.) to convey steganographic data (e.g., Bedoop data). When such a note is presented to a Bedoop system, the system may launch an application that stores a snapshot of the note. More particularly, the application may mask the note-portion of the image data from the other image data, virtually re-map it to a rectangular format of standardized pixel dimensions, JPEG-compress the resulting image, and store it in a particular computer subdirectory with a name indicating the date of image acquisition, together with the color and/or size of the note. (These latter two data may be indicated by data included in the Bedoop payload.) If the color of the note is indicated by digital data (e.g., in the file name), then the image itself may be stored in grey-scale. When later recalled for display, the white image background can be flooded with color in accordance with the digital color data.

The Bedoop system may buffer several past frames of image data. When the object is recognized as a Post-It note whose image is to be saved, the system may analyze several such frames to identify the one best-suited for storage (e.g., check the spatial frequency content of the note as imaged in each frame, to identify the one with the finest detail), and store that one.

When a Post-It note is recognized by the Bedoop system, the system may emit a confirmation tone (or other response) to indicate that the object has been recognized, but not immediately execute the snapshot operation. Instead, the system may await a further instruction (e.g., gesture) to indicate what operation is desired.

By moving the note towards the sensor, for example, the user can signal that a snapshot operation is to be performed. (This closer presentation of the note may also permit the imaging system to capture a more detailed frame of image data.)

By moving the note away, the system may respond by reading, decompressing, and displaying the six most-recently stored Post-It note images, in tiled fashion, on the computer screen. The individual notes can be displayed at their original dimensions, or each can be re-sized to fill the full height or width of a tile. A user interface control (responsive to gestures, mouse operation, keyboard scroll arrows, etc.) allows the user to scroll back in time to any desired date.

The full 64-bit Bedoop payload of other embodiments may not be needed for Post-It notes. In the just-given example, for example, the Bedoop system responds to all Post-It notes in the same fashion. Thus, an abbreviated Bedoop format that indicates simply 'I'm a Post-It note, yellow, size 3" x 3"' can suffice. The twelve bit CLASS ID, with eight further bits to indicate color/size combinations, may be

sufficient. Reducing the payload permits it to be more robustly encoded on small objects. (As noted below, Bedoop decoding systems can look for several different data formats/protocols in trying to extract Bedoop data from an object.)

Alignment of Documents for Other Purposes

While the just-described pre-marked paper triggered a Bedoop response when presented to a Bedoop sensor (i.e., take a snapshot of the paper), the markings can be used for purposes other than to trigger Bedoop responses.

Regardless of the particular data with which the paper is encoded, the embedded subliminal graticules, or other steganographically-encoded registration data, can be used by other applications to correct misalignment of scanned data. In a photocopier, for example, a document need not be placed exactly squarely on the glass platen in order to yield a properly-aligned photocopy. The scanner scans the skewed document and then detects the steganographic registration markings in the resulting scan data. This data is then processed to virtually re-register same, so that the registration markings are in a desired alignment. The processed scan data is then provided to the xerographic reproduction unit to yield a photocopy in which the skew effect is removed.

The same technique is likewise applicable to video recorders, digital cameras, etc. If such a device images an object (e.g., a photograph) with steganographic registration markings, these markings can be used as a guide in re-registering the resulting data to remove mis-alignment effects.

Postal Mail Information

Many contexts arise in which data to be presented to a consumer is valuable only if timely. The postal service mail is ill-suited for some such information due to the latency between printing a document, and its ultimate delivery to a recipient. Bedoop principles, however, allow the recipient to take a postal object that was printed well before delivery, and use it on receipt (i.e., present to a Bedoop system) to receive up-to-the-minute information. In this and other embodiments, the Bedoop data can also uniquely identify the addressee/recipient/user, so the web site can present data customized to that user.

Distributors of printed advertising can reward Bedoop-driven consumer visits to their web sites by issuing digital tokens or coupons that can be redeemed for premiums, cash-back, etc. Every millionth visitor wins a million pennies (with appropriate safeguards, e.g., preventing more than one entry an hour).

Classes of Bedoop Encoding

The above-described embodiments focused on use of Bedoop data after decoding. Additional insight may be gained by examining the earlier part of the process – encoding.

Encoding can be performed in many contexts, which may be conceptualized as falling into three broad classes. The first is static marking, in which a document designer, pre-press service bureau,

advertising agency or the like embeds Bedoop data. The second is dynamic marking, in which automated systems encode, or vary, Bedoop data "on the fly." Such systems can tailor the Bedoop data to particularly suit the context, e.g., to the moment, place, user, etc. The third is consumer marking, in which Bedoop data is added to a document at the time of printing.

The second class of encoding enables features not available from the first. Consider an American Express travel web page with information about travel to Hawaii. A DNS leaf node server points to this page in response to certain Bedoop data – e.g., data encoded in a magazine photograph of a Hawaiian beach scene.

Actually, all Bedoop data having a certain CLASS and DNS ID may lead to this web page, irrespective of the UID data. If the magazine photo is encoded with a particular "don't care" UID field (e.g., 111111111111111111111111), this may signal the originating Bedoop system – or any intervening system through which the Bedoop data passes – that arbitrary data can be inserted in the UID field of that Bedoop packet. The originating Bedoop system, for example, can insert a dynamically-configured series of bits into this field. Some of these bits can provide a profile of the user to the remote server, so that the Bedoop response can be customized to the user. (The user would naturally pre-approve information for such use so as to allay privacy concerns.)

As one example, the local Bedoop system can set the least significant bit of the UID field to a "0" if the user is male, or to a "1" if the user is female. The next four bits can indicate the user's age by one of sixteen age ranges (e.g., 3 or less, 4-5, 6-7, 8-9, 10-11, 12-13, 14-15, 16-17, 18-20, 21-24, etc.).

Alternatively, or in addition, the local Bedoop system can stuff the don't-care UID field (all of it, or in part) with signature data tending to uniquely identify the local Bedoop system (e.g., system serial number, a hash code based on unchanging data unique to that system, etc.) By reference to such data, the remote server can identify repeat visits by the same user, and can tailor its responses accordingly (e.g., by recalling a profile of information earlier entered by the user and stored at the remote server, avoiding the need for data re-entry).

More on Optical Input Devices

It is expected that image input devices will soon become commonplace. The provision of digital cameras as built-in components of certain computers (e.g., the Sony Vaio laptops) is just one manifestation of this trend. Another is camera-on-a-chip systems, as typified by U.S. Patent 5,841,126 and detailed in Nixon et al., "256x256 CMOS Active Pixel Sensor Camera-on-a-Chip," IEEE J. Solid-State Circuits, Vol. 31(12), pp. 2046-2051 (1996), and Fossum, "CMOS Image Sensors: Electronic Camera-on-a-Chip," IEEE Transactions of Electron Devices, vol. 44, No. 10, Oct. 1997. Still another is head-mounted cameras (as are presently used in some computer-augmented vision systems). These and other image input devices are all suitable for use in Bedoop systems.

Camera-on-a-chip systems can be equipped with Bedoop detector hardware integrated on the same chip substrate. This hardware can be arranged to find and decode Bedoop data from the image data – notwithstanding scale, rotation, differential scaling, etc. Gestural decoding can also be provided in

hardware, with the resulting data output in packet form on a serial output bus. Such a chip can thus provide several outputs – image data (either in raw pixel form, or in a data stream representing the image in one of various image formats), 64 bits of Bedoop data (serially or in parallel), and decoded gesture data.

In other embodiments, the Bedoop detector (and/or the gestural decoder) can be on a substrate separate from the camera system.

To accommodate different Bedoop data formats and protocols, the hardware can include RAM or ROM in which different format/protocol information is stored. (These different formats/protocols can relate, e.g., to Bedoop systems employing different data payload lengths, different subliminal grids, different encoding techniques, etc.) As the Bedoop system stares out and grabs/analyzes frames, each frame can be analyzed in accordance with several different formats/protocols to try and find a format/protocol that yields valid Bedoop output data.

Movable Bedoop Sensors

Although the illustrated Bedoop systems are generally stationary, they need not be so. They can be portable. Some such systems, for example, employ palmtop computers equipped with optical sensor arrays. If the palmtop is provided with live network connectivity (e.g., by wireless), then Bedoop applications that rely on remote computers can be implemented just as described. If the palmtop is not equipped with live network connectivity, any Bedoop applications that rely on remote computers can simply queue such communications, and dispatch same when the palmtop next has remote access (e.g., when the palmtop is next placed in its recharger and is coupled to a modem through which internet access can be established).

Another variant is a Bedoop sensor with a 1D or 2D photosensor array (e.g., CCD or CMOS) that is movable around a desk or other work-surface, like a mouse. Such a sensor can be coupled to the associated computer by cabling, or a wireless interface can be used. The peripheral may be arranged for placement on top of an item in order to read digital data with which the object is marked. (Built-in illumination may be needed, since the device would likely shadow the encoding.) Some forms of such peripherals are adapted to serve both as general purpose digital cameras, and also as Bedoop sensors.

Such a peripheral would find many applications. In "reading" a magazine or book, for example, it may be more intuitive to place a Bedoop reader "on" the object being read, rather than holding the object in the air, in front of a Bedoop sensor. This is particularly useful, e.g., when a magazine page or the like may have several differently-encoded Bedoop sections (corresponding to different articles, advertisements, etc.), and the user wants to assure that the desired Bedoop-encoded section is read.

The "bookmark" paradigm of internet browsers might be supplemented with paper bookmarks, e.g., Bedoop data encoded on one or more pages of paper. To direct a browser to a particular bookmarked destination, the peripheral is simply placed on top of the page (or part thereof) that is marked with the corresponding Bedoop data. A user may print a "map" comprised of postage stamp-sized regions tiled together, each of which regions represents a favorite web destination.

Such a map may be printed on a mouse pad. Indeed, mouse pads with certain maps pre-encoded thereon may be suitable as promotional materials. A company may offer to print a family photograph on such a pad. Encoded within the photograph or the pad texture are addresses of web sites that have paid a fee to be accessible in this manner on a user's desk.

In this and other contexts, it will be recognized that the gestural input concepts detailed earlier involve relative movement between the sensor and the encoded object. In most of the earlier examples, the sensor is stationary, so gesticulation is effected by moving the object. Naturally, if the sensor is moveable (e.g., as a mouse or cell phone), the gestural movement can be effected by moving the sensor instead.

One particular embodiment of the arrangement detailed above a modified version of the Microsoft IntelliMouse with IntelliEye technology. That device includes a multi-element optical sensor integrated on an IC with various detector and processing circuitry, operating in conjunction with a short focal length imaging lens and an LED illumination source (all available from Agilent, as detailed below). The circuitry tracks movement of patterns across the sensor's field of view, by which the mouse's movement can be deduced. The Microsoft product collects 1500 data sets per second – a frame rate much higher than is generally needed for the present applications. Some such embodiments combine the functionality of a mouse with that of a Bedoop image sensor.

Such a mouse-like peripheral can omit the buttons and position-sensing features commonly provided on traditional mice, yielding a simple desk-facing palm camera that generates frames of data corresponding to a small area under the sensor portion of the mouse. More typically, however, the peripheral includes the buttons, roller wheels, and/or X-/Y- position sensing arrangements of traditional mice so that button and positional forms of data input can be exploited in interacting with the Bedoop application.

The optical data collected by the sensor can be processed within the peripheral's processor to extract the steganographically encoded binary Bedoop data therefrom. Or this processing burden can be undertaken by the associated computer system, with the peripheral simply processing and formatting the raw sensor data into sequential frames of image data to be output to that system.

While scanning peripherals of the type described above are typically wired to an associated host system, wireless links (e.g., radio, infrared, ultrasonic, etc.) can of course be used, freeing the user from the constraint imposed by the cable.

Hand Scanners – More

To elaborate on the concepts introduced above, the following paragraphs particularly detail a hand scanner modeled after the Hewlett-Packard CapShare 920. The scanner may be configured for use with any type of identifier, e.g., watermark, barcode, OCR, etc.

The reader is presumed familiar with the workings of the HP CapShare scanner. Such information has been published in the technical literature, e.g., Allen, Ross R. et al, "Processes for Freehand Image Capture: HP CapShare Technology," The Society for Imaging Science & Technology

PICS Conference, pp. 43-46, March 26, 2000. The CapShare scanner employs an optical sensor tracking engine assembly, comprised of part Nos. HDNS2000 (IC sensor chip), HDNS-2100 (lens with light pipe), HDNS-2200 (LED clip), and HLMP-ED80 (639 nm LED), the use of which is detailed in Agilent Application Note 1179, all available from Agilent Technologies, Palo Alto, CA.

As shown in Fig. 13, the CapShare scanner 210 functionally includes a CPU 212, a memory 214, a linear CCD array 216, first and second 2D CCD arrays 218a, 218b, a user interface (including an LCD screen and associated buttons) 220, and an interface 222 (e.g., a serial port and an IRDA port/transceiver) to an auxiliary device 224. Although not particularly shown, each of the CCD sensors has an associated LED system for illuminating the object being imaged.

A further enhancement to the CapShare scanner is to provide a wireless internet interface for interface 222, permitting direct communication between the device 210 and the internet.

The linear sensor 216 operates in a conventional manner, acquiring successive pixilated line-scans of imagery under the scanner at a pixel data rate of about 18 Mbits/second (permitting full accuracy scanning even at instantaneous scanner motion speeds of up to 16 inches per second).

The two 2D CCDs 218a, 218b, are spaced apart, adjacent the linear sensor, and are used to track the scanner's movement. The areas scanned by these CCDs are illuminated obliquely by IR light, highlighting microscopic media surface features. The CPU identifies patterns in the surface features detected by each of these CCDs, and tracks their movement from one frame to the next to discern the movement of the two CCDs. By knowing the movement of the two CCDs, the movement of the scanner itself can be determined. This scanner motion information is used to re-map the swathed scan data acquired from the linear scanner array into composite pixel data with uniform scanline spacing. This remapped data is the final scan data that is typically provided to the end user or application.

The sizing of the linear and 2D CCD arrays is left to the designer. If 600 dpi line scan resolution is desired across a four inch scan swath, a 2400 element linear CCD would be used. The 2D CCDs are typically smaller in extent, e.g., having an aperture of 0.25 inches x 0.25 inches. The spacing of the CCD elements is selected based on the size of surface features to be tracked.

The memory 214 includes both RAM and ROM. The RAM is used both for raw data storage, and for storage of final results. The scanner's operating system, pattern matching and data processing algorithms, and other program code are stored in the ROM.

In accordance with the illustrated embodiment, the ROM code also includes instructions for examining the final-result scan data (i.e., after re-mapping) for watermark data. In one particular embodiment, this is a two-step process. The final-result data is first screened by the CPU to determine whether it has certain hallmarks associated with watermarked data (e.g., the presence of a calibration signal, as detailed in U.S. patent 5,862,260). If such hallmarks are found, a more computationally-intense watermark decoding algorithm is performed. By checking for watermark information in such a two-step process, CPU time isn't spent needlessly trying to extract a watermark from image data that apparently has no watermark.

In some embodiments, the watermark processing operation(s) occurs without user intervention – each time the raw scan data is processed and remapped into final-result form. In other embodiments, the watermark processing is invoked in response to a user command entered through the user interface 220. In still other embodiments, the watermark processing is invoked in response to a command provided to the scanner from an associated auxiliary device 224 (e.g., a local personal computer, a remote server computer, a specialized internet appliance, etc.).

In some embodiments, the decoded watermark payload data is transferred to an associated auxiliary device whenever such data is detected. In other embodiments, the data is not sent unless the auxiliary device 224 first asks for it. Such an auxiliary device request may be made by a software program that is executing on the auxiliary device, e.g., through an API call.

A watermark decoded by the scanner can be presented to the user on the scanner's LCD display 220. Or the scanner can forward the decoded watermark to a remote device 224, which can then reply with supplemental data for presentation to the user, as detailed elsewhere in this specification.

In some embodiments, the information obtained by the 2D CCDs 218a and/or 218b can be used to augment the information provided to the CPU so as to aid in the watermark detection/decoding process. For example, the 2D sensors provide 2D information immediately – without the time delay associated with remapping the 1D data from the linear CCD array into final form. This immediate 2D information may be analyzed for the presence of a calibration signal – perhaps shortening or obviating analysis for the calibration signal in the final data.

The 2D data can also be used to characterize the texture of the imaged substrate. Knowing the substrate texture allows the CPU to apply filtering or other watermark-enhancing/image enhancing steps as best fits the particular circumstance.

The 2D data also allows the CPU to assess the relative quality of different regions of line-scan data in the final scan data. In Fig. 14, an object 230 is scanned by a scanner 210 traveling an arced path, yielding successive lines of raw scan data 228. The geometry of the arc is revealed by data collected by the two 2D CCDs (which serve as motion encoders). Where the successive lines are spaced closer together (as at region A), the raw scan data is of higher resolution (relatively "oversampled"). In contrast, where the successive lines are spaced further apart (as at region B), the raw scan data is of lower resolution (relatively "undersampled"). In its screening or analysis for subtle watermark data, the CPU can be instructed to look first in region A, reasoning that the higher quality of the raw data in this region is most likely to include usable watermark information. (In some embodiments, watermark screening and/or detection may proceed by reference to the raw scan line data, rather than waiting for the data to be remapped into final form.)

The two CCDs 218a, 218b also permit various binocular processing to be performed so as to enhance, or allow for more intelligent processing, of the linear scan data. Consider, for example, the case where the image being scanned doesn't convey a perfectly planar (e.g., rectilinear) signal. The substrate could be warped, or the image may have been printed in a skewed manner, or some other transformation may have taken place that creates an image that changes orientation/scale over its extent. One such

example is shown in Fig. 15, which depicts a watermark calibration signal (here presented as an overt grid, for clarity of illustration). Each of the two 2D CCDs 218a, 218b senses a differently-oriented calibration signal. By reference to differences in the data sensed by the two spaced-apart CCDs, the CPU can infer scale, rotation, or other image transformation at intermediate points, and apply a compensation to the scan data so as to counteract such distortion.

Such compensation need not be based on a watermark calibration signal. Any other optically-sensed attribute at the two spaced-apart points may be used to infer the same, or related attributes, at intermediate positions, permitting appropriate compensation(s) to be applied.

Another application of the spaced-apart sensors 218 is to bi-optically determine the distance from the scanner to the object being imaged. Consider Fig. 16, which shows a scanner 10 with two spaced-apart CCD sensors 218a, 218b, imaging a warped object 232 (e.g., a warped paper substrate, or a curved object such as a drink can). In the depicted example, the first CCD 218a detects a very crisp image of surface texture or imagery, whereas the second CCD 218b detects an image similar to the first CCD, but blurred – as if low-pass filtered. Software instructions for CPU 212 can cause the scanner to recognize that the part of the object 232 under CCD 218a is immediately adjacent the scanner (and adjacent the nearby portion of the linear CCD array 216), whereas the part of the object under CCD 218b is spaced somewhat from the scanner.

Having information about the spacing of the object 232 from different parts of the linear sensor array permits certain compensations to be made in the data collected by the linear scanner. In the case of watermark detection, the CPU may recognize that data from the end of the linear scanner 216 remote from the object 232 will be devoid of meaningful high frequency information. In many watermark detection algorithms, the CPU can disregard data from such portion of the linear scanner – allowing it to focus on portions of the image data with the necessary high frequency components, thus producing more reliable results more quickly.

In other watermarking systems, two watermarks may be present – one conveyed in reliance on high frequency image components, and another encoded mostly with low frequency image components (i.e., in anticipation of circumstances where the object is imaged slightly out-of-focus). Upon recognizing that the linear scan data is weak in high frequency components, the CPU can be programmed to look just for the low frequency watermark data.

Just as the scanner 210 can perform watermark decoding, it can similarly perform barcode decoding. Conventional pattern-recognition algorithms can be applied by CPU 212 to either the raw or final scan data to identify barcode patterns. Once the pattern is identified, decoding is straightforward by applying known barcode alphabets. As in the watermark case, barcode decoding can be performed autonomously, or in response to user/auxiliary device command. Similarly, the decoded barcode data can be provided to the auxiliary device whenever detected, or in response to an auxiliary device query. The 2D data can likewise be used to augment the information provided to the CPU so as to aid in the barcode detection/decoding process.

In similar fashion, the above-described benefits associated by use of the two 2D CCDs can similarly be applied in the barcode context.

As earlier described, once the identifier information is extracted from the image data, internet links can be based thereon to provide supplemental information, e-commerce opportunities, etc. In many implementations, the scanner UI 220 is used to present this supplemental information to the user, e.g. by software instructions that render HTML instructions for presentation on the UI display screen. The UI controls (e.g., buttons) can likewise be used to receive user instructions and commands, for linking back to the internet.

Cell Phones as Bedoop Devices; GPS Receivers

Bedoop technology can be integrated into portable telecommunication terminals, such as cell phones (manufactured, e.g., by Motorola, Nokia, Qualcomm, and others). Such a phone can be equipped with a 1D or 2D image sensor, the output of which is applied to Bedoop decoding circuitry within the phone. This decoding circuitry can be the phone's main CPU, or can be a processing circuit dedicated to Bedoop functionality. (In this as in other embodiments, , the decoding can be effected by dedicated hardware, by decoding software executing on a general purpose CPU, etc.)

Cell phones are already equipped with numerous features that make them well suited for Bedoop operation. One is that cell phones typically include an LCD or similar screen for display of textual or graphic information, and additionally include buttons or other controls for selecting among menu options presented on the screen (e.g., by moving a cursor). Moreover, cell phones naturally include both audio input and output devices (i.e., microphone and speaker). Still further, the protocol by which cell phones transmit data includes data identifying the phone, so that such data need not be separately encoded. And finally, cell phones obviously provide ready links to remote computer systems. Collectively, these capabilities rival those of the most fully-equipped desktop computer system. Thus, essentially all of the applications detailed elsewhere in this specification can be implemented using cell phone Bedoop systems.

As with the other Bedoop systems, when Bedoop data is sensed, the phone can respond to the data locally, or it can forward same over the cellular network to a remote system (or computer network) for handling.

One application that may be invoked locally (i.e., within the phone) is the dialing of a corresponding telephone number. In some embodiments, the phone number is literally encoded as part of the Bedoop data payload. In others, the phone number is stored in a memory within the phone, and indexed in accordance with an identifier decoded from the Bedoop data.

The variety of operations that can be handled remotely is virtually limitless. Some entail interaction with the user. For example, the remote system may initially respond by presenting to the user a menu of options on the display screen (e.g., Purchase, Add to Shopping List, Request Sample, Add to Notepad, etc.) The user then responds by providing further input (e.g., by manipulating one or more buttons on the phone, by providing spoken instructions to a voice recognition sub-system within the

phone, etc.). This further data is then dispatched from the phone, and the requested action undertaken. Other operations don't require further interaction with the user, but immediately invoke a corresponding action.

While the just-described arrangement included the Bedoop decoding function within the phone, in other embodiments the image data can be transmitted from the phone and decoded at a remote location.

Reference was earlier made to GPS receivers as permitting the location of a person to be tracked, and contact information updated accordingly. GPS receivers find many other applications in Bedoop contexts. For example, the response of a Bedoop system can be tailored, or vary, in accordance with the location of the person invoking the operation. To illustrate, if a user presents a newspaper insert or coupon for a Dominos pepperoni pizza meal to the Bedoop sensor on a GPS-equipped cellular phone, the GPS data can be encoded in the Bedoop data dispatched to the Domino's Bedoop server. That server can identify, by reference to the GPS data, the location of the nearest Domino's franchise, and can forward the order to that franchisee. The franchisee, in turn, can telephone the user (by reference to telephone number data from the cell phone) to confirm the order, inquire if additional items are desired, inform the user of the final price, and to obtain the delivery address. (The latter step can be omitted; the franchisee can employ the GPS data to obtain the corresponding street address, e.g., from map data licensed through NavTech of Chicago.)

The protocols by which the Bedoop data, GPS data, and cell phone identification data are conveyed from the phone to the cellular network can take various forms; the design of such systems are familiar to those in the relevant arts. In one embodiment, the protocols by which some cell phones are now provided with email or internet access are further adapted to convey Bedoop and GPS data. The protocols for conveying telephone identification data are already well established. Once received by the cellular network, the Bedoop data can be formatted into packets, either incorporating the GPS data and telephone data into the same packet structure, or by formatting such data into separate packets accompanying the Bedoop packets.

The provision of image sensors in cell phones enables other functionality. One is the capture of still or video imagery. Such image data can be compressed (typically by lossy processes such as MPEG, JPEG, or the like, implemented with dedicated hardware CODECs) and transmitted with the audio data. The screens on such phones can likewise be used for display of incoming image or video data.

Another function enabled by image sensors in cell phones is user-verification, e.g., by retinal scanning or other optically-sensed biometrics, before the phone will permit a call to be placed. A great number of such biometric verification techniques are known.

Cell phone Bedoop sensors may not always be in communication with the cell phone network. The phone may be out of range of a cell site, or may be in operational mode in which an RF link is not then established. In such events, any Bedoop data sensed by the phone that is to be handled remotely is desirably stored locally within the phone, and queued for transmission to the cellular network the next time communication is established (a so-called "store and forward" form of operation).

Catalog Advertising

Any form of hand-held scanner - whether of the type just described or others known in the art - offers a convenient way to interact with catalog advertising. Imagine a traditional paper catalog, e.g., from L.L. Bean, Inc., or Lands End. Each image in the catalog is Bedoop-encoded with a code that identifies the depicted product (and, if necessary, the manufacturer, etc.). A user browsing through the catalog, on seeing a product of interest, places the scanner over the picture (and optionally may be required to push a button or otherwise signal to initiate further processing). The scanner detects the Bedoop data and relays it to an associated computer (optionally with data identifying the consumer). The computer polls a remote server computer maintained by the merchant, which responds with data corresponding to the item depicted in the scanned image. This returned data can include data indicating the sizes available, data indicating the colors available, data indicating the variant styles available, flag bits indicating whether each item is in stock, etc. This returned data can be presented to the consumer – typically on a display device but alternatively in audible form.

Preferably, the customer's body measurements (waist size, inseam length, neck size, etc.) are stored in a user profile, either on the local computer, at the merchant's server computer, or at the computer of a third party service provider. This allows the system to customize the data presented to the user – e.g., showing the color options and availability only for the depicted shirt in a 16 inch neck and a 34 inch sleeve length.

If necessary, the user can select among the color or style options, using the handheld input device (either buttons, gestures, etc.), or any other input device. Or the item may be one for which no further specifications are needed. In either event, once the desired product has been sufficiently specified, the user can signal the system to place the order. Payment and shipping details can be arranged through any of the great variety of techniques known in the art, e.g., by charging to a credit card number and shipping to an address on-file with the merchant.

Revenue Sharing

When a consumer presents a Bedoop-encoded object to a sensor, and as a result of the link(s) thereby established, purchases a product or service, the revenue from that transaction may be shared with the participants who made it possible. In the case of a Bedoop-encoded magazine ad, some of the participants may include (1) the photographer or graphic designer who produced artwork used in the ad; (2) the advertising agency whose creative talent led to the ad; (3) the publisher of the magazine in which the consumer encountered the ad; (4) the service provider(s) who provided the transport channel(s) between the consumer and the vendor; and (5) the service provider who maintained the server that ultimately linked the Bedoop data to the vendor's web site.

Gambling Applications

Casinos around the world deploy considerable resources each year to deter cheating. To that end, playing cards and chips used for gaming and betting could be encoded with digital watermarks and used to enhance security. In order for the application to work, playing cards and chips would be encoded with data unique to each casino, game, card deck and/or face value. Then, cameras placed above gaming tables would be used to read the information off the cards and chips. The information from the cards could be used to alert security or keep records of suspicious activity in the following ways.

A camera placed above a card gaming table (like Blackjack) could read the data encoded in the backs of the cards determining the authenticity and face value of those cards. This means the camera could catch anyone trying to replace cards after they were dealt-either by flagging cards as not being part of the deck currently in use or by flagging them as being a different face value than those dealt to the player. If a player switches cards after they are dealt, the watermark reader in the camera can alert security of the fraudulent activity.

Additionally, the data encoded in the cards can be used to track the winner(s) at the table. If the dealer and the player are in collusion, that player may be dealt winning cards on a regular, predetermined basis (based on tricky shuffling by the dealer). By watching the pattern and face value of cards that are dealt, it can be determined if a player wins at a much higher than average rate. Again, security personnel can be alerted to suspicious activity.

Cameras placed above gaming tables where chips are used (certainly not limited to card games) can be programmed to watch for watermarks in chips. Any chip that does not include the appropriate watermark can signal that a chip might be counterfeit. In another situation, the face value of the chips can be determined as they are placed on the table such that any switching of the chips after play begins will be noted.

The foregoing principles are more generally applicable to monitoring and surveillance systems employing digital watermark technology to identify monitored objects.

PART II

As noted, the following disclosure focuses on one particular application – a system for linking print media to electronic content. It should be reiterated, however, that the technology is not so limited, and may more generally be viewed as a system for linking any object (physical or electronic) to a corresponding networked or local resource.

In accordance with an exemplary application of the below-detailed technology, digital watermarking is employed to convey a plural bit identifier within print media, such as magazine advertisements or articles, direct mail coupons or catalogs, bank- or credit-cards, and business cards. This identifier is read by software in a user's computing device and forwarded to a remote database. The remote database identifies a URL corresponding to the identifier, and provides the URL back to the user's computer – permitting a browser on the user's computer to display the URL-identified web page. That web page can provide additional information or services – more timely and/or more extensive than

that provided by the print material. By such arrangement, more efficient internet navigation and access is provided to consumers, and more effective means for linking readers to e-commerce points of sale is provided to advertisers.

Before beginning a detailed exposition, it may be helpful to provide an overview of the larger system of which the present technology forms a part. As shown in Fig. 1, the larger system entails four basic processes – registering, embedding, detection and response.

Registering refers to the process of assigning an ID to an object, and associating that ID with a corresponding action or response. Additional steps can be included, such as logging the name and/or organization of the registrant, the name of the product, a description of the object and a context in which it is found (magazine, book, audio track, etc.), etc.

Embedding refers to the process of encoding of an object with a digital identifier (e.g., a watermark conveying a serial number in its payload).

Detection is the complementary operation to embedding, i.e., discerning a digital identifier from an object.

Response refers to the action taken based on the discerned identifier.

The middle two steps – embedding and detection – can employ any of myriad well-known technologies, including 1D and 2D barcodes, magnetic ink character recognition (MICR), optical character recognition (OCR), optical mark recognition (OMR), radio frequency identification (RF/ID), data glyphs, organic transistors, magnetic stripe, metadata, file header information, UV/IR identifiers, and other machine-readable indicia and techniques for associating plural-bit digital data with an electronic or physical object. The detailed embodiment employs watermarking technology, although this is illustrative only.

Referring to Fig. 2, a system 10 according to the exemplary embodiment includes an originating device 12, a router/server 14, a product handler 16, a registration database 17, and one or more remote resources 18.

The originating device 12 can take many different forms, e.g., a cell phone, a personal digital assistant (e.g., a Palm Pilot), a personal computer, a barcode scanning system, etc. For expository convenience, the embodiment is described with reference to a personal computer for device 12.

Device 12 interacts with an object 20. The object can be electronic or not. Electronic objects 20 can include computer files, representations of audio, video, or still imagery (e.g., files or in streaming form), etc. Non-electronic objects can include physical objects such as newspapers, magazine pages, posters, product packaging, event tickets, credit cards, paper currency, etc. Non-electronic objects can also include sounds produced by loudspeakers.

When used with non-electronic objects, device 12 (Fig. 2) typically includes some form of sensor or transducer 22 to produce electronic signals or data corresponding to the object. Examples include CCD- or CMOS-based optical sensors (either as part of still- or video cameras, flatbed scanners, mice, or otherwise), microphones, barcode scanners, RF ID sensors, mag stripe readers, etc. In such cases, the sensor 22 may be coupled to associated interface electronics 24, which in turn may be coupled to device driver software 26, which in turn may be coupled to one or more application programs 28.

Device driver software 26 serves as a software interface, communicating at a relatively high level with the application programs 28 (e.g., through API instructions whose content and format are standardized to facilitate application programming), and at a relatively low level with the interface electronics 24.

The detailed embodiment contemplates that the object 20 is a magazine advertisement encoded with a steganographic watermark conveying a plural bit object identifier. The watermark is hidden in the advertisement's image in a manner indiscernable to human observers, but detectable by computer analysis. That analysis is performed by a watermark detector 30.

Watermark detector 30 can be implemented in various different locations in the system of Fig. 1. Typically, the detector is implemented in the originating device 12, e.g., in the driver software 26, or in application software 28c that serves to link to external resources based on detected watermarks. But it may be implemented elsewhere, e.g., in hardware in the interface electronics 24, in an operating system associated with the device, or outside device 12 altogether. Some systems may have plural watermark detectors, implemented at different locations throughout the system.

In an illustrative system, the watermark detector is implemented in the device driver 26. Functionality of the detector is made available to the application program 28c through one or more APIs specific to watermark-related functions. One function is reading of the watermark data payload from the object 20.

The illustrated application 28c is a software program that serves to communicate the watermark data from the device 12 to the router/server 14 through one or more communications links 32 (e.g., the internet). Application 28c also receives information from the communication links 32 and presents same to the user (or otherwise uses same).

The router/server 14 is a high capacity computer including one or more CPUs, memory, disks, and I/O ports. As is familiar to artisans, the disks store operating system software and application programs, together with data, that are transferred to the memory as needed by the CPU. The router essentially serves as a middleman between the application 28c and the product handler 16. As detailed below, the router receives requests from the application, logs them in a transaction log 15, and passes them on to the appropriate product handler.

As more particularly detailed below, the handler 16 provides a response in accordance with the particular watermark payload. The response may be provided directly by the product handler to the device 12, or the handler may respond by communicating with a remote resource 18 (which may be, e.g., a data repository or service provider).

In the former case, the handler 16 may identify a URL corresponding to the watermark (using the database 17), and return the URL to the application 28c. Application 28c can then pass the URL to a web browser 28b in the device 12, and initiate a link to the internet site identified by the URL. Or the handler may have some locally stored data (e.g., audio or video, or software updates) and send it to the device 12 in response to the watermark.

In the latter case, the handler 16 does not respond directly to the device 12. Instead, the handler responds by communicating with a remote resource 18. The communication can be as simple as logging

receipt of the watermark message in a remote repository. Or it can be to authenticate device 12 (or a user thereof) to a remote resource in anticipation of a further transaction (e.g., the communication can form part of an on-line licensing or digital rights management transaction). Or the communication can request the remote resource to provide data or a service back to device 12 or to another destination (e.g., to initiate an FTP file transfer, or to request that a song selection identified by the watermark be downloaded to a user's personal music library, or to update software installed on device 12).

In still other cases, hybrids of the two foregoing cases can be employed, e.g., handler 16 can send some data back to device 12, while also communicating with a remote resource 18.

In some cases, the response returned to the device 12 by handler 16 (or a remote resource 18) can serve to trigger some further action by the device 12. For example, the response returned to device 12 can include a WindowsMedia audio file, together with a request that the device 12 launch the WindowsMedia player installed on the device. (The launching of a browser pointed to a URL is another example of such triggering.)

The illustrated product handler 16 comprises essentially the same hardware elements as the router 14, e.g., CPU, memory, etc. Although Fig. 2 shows just one product handler, several product handlers can be included in the system – either co-located of geographically distributed. Different handlers can be dedicated to different functions (e.g., serving URLs, serving music, etc.) or to different watermark sources (e.g., one responds to watermarks found in audio, another responds to watermarks found in print advertising, etc.). Further specialization may also be desirable (e.g., one handler may respond to advertising placed by Ford, another may respond to advertising placed by Chevrolet; or one handler may respond to advertising appearing in Wired magazine, another may respond to advertising appearing in Time magazine, etc.). In one particular implementation, the router 14 dispatches the incoming data to one of several handlers in accordance with (1) the vendor of the originating application 28c, and (2) the particular identity of the application 28c.

The following discussion focuses on the data exchanged between the application 28c, the router/server 14, the product handler 16, and the associated protocols, in one illustrative embodiment.

Concept of Operation

When shown a watermarked image, the application 28c analyzes the image and extracts the embedded watermark payload (more particularly detailed below) from the image. The application sends some or all of this information in a message format to the router 14.

The router 14 decodes the received message, looking for vendor and product information. Based on this information, it passes the message to a corresponding product handler 16.

The product handler receives the message and attempts to match the detected watermark serial number to a registered watermark serial number earlier stored in the database 17. If a match is found, the product handler performs the desired action. As noted, typical actions include returning a URL for web redirection, serving up an HTML page for initial user navigation, initiating software downloads, etc. If a match is not found, the product handler returns an error code and message to the application 28c. If a

match is found, but the corresponding action is unavailable, incomplete, inactive or invalid, the product handler returns an error code and message to the calling application.

A generalized view of the foregoing is provided in Fig. 4.

(Note that while the system may concentrate on a certain type of object 20, and a certain vendor's application 28c, the architecture is constructed to support accessing product handlers from other vendors and corresponding to other objects. This concept makes the system suitable as a clearinghouse for processing all machine-readable indicia on web-enabled devices.)

An exemplary detection and response cycle is illustrated below.

User	Application	Router	Product Handler
Shows object to sensor 22			
	Acquires watermark Creates message packet Sends packet to product handler		
		Receives message packet Logs transaction Decodes packet Identifies product sending packet Passes packet to Product Handler corresponding to product	
			Logs received packet Validates packet serial number If not found, returns error packet to application Else, returns packet with data/action back to application (e.g., URL)
	Receives packet If error, display error message Else, display data or perform the requested action (e.g., launch browser and link based on received URL)		
Sees the data/action associated with the object (e.g., views web page)			

The present system generalizes this example to support any product from any vendor that is capable of sending a message via the Internet that complies with the expected request format (e.g., a product code, message type, and identifier) and receiving a message in a corresponding response format. One set of message formats suitable for use in such a system are described in more detail below.

Watermark Registration – the first step in the process

In order for the system to identify the response (e.g., a URL) that corresponds to an object identifier (e.g., a watermark), this data must first be associated within the database 17 in association with the watermark to which it corresponds. The watermark registration process captures some basic identification information used later to validate the incoming message, and identifies the associated information/action. In the illustrated example the identification information includes:

- Customer Account,
- Object and associated attributes (name, description, expiration, etc.),
- Action, and
- Registered Serial Number (for registration updates)

The Customer Account identifies the watermark registrant. In most cases, this is also the party to be billed for services. For validation and security reasons, the Customer Account is required to be a known, existing account. Account information, including the account's password, is maintained by an Account Management system.

The Object and associated attributes identifies the object to be watermarked. The object attributes typically include the name and description of the object and a list of accounts authorized to access the object's registration. These authorized "supporting" accounts are typically the ad agencies, pre-press houses, etc. involved in the watermark embedding process in the print advertising example contemplated herein.

The Action defines the response the customer desires when the watermark is detected. It varies by product, but in the illustrative embodiment involves the return of some additional information regarding the watermarked object. In the illustrative system, the action is return of a URL or HTML to be used to display a web page associated with the watermarked object. For other products, the desired response may be display of the object's owner & rights information, software/data downloads, delivery of streamed audio or video, presentation of an advertisement, initiation of object-based actions, etc.

The Registered Serial Number forms the last component of the registration. It is this assigned vendor and product-unique identifier that allows the system to acquire the specific information/action for the object in question.

A few key product Registration concepts -

Watermark Registration is a product-specific process –

To allow each of the products the freedom to upgrade their capabilities without impacting any other product function or schedule, the registration process is product-specific.

Watermark registration is web-enabled -

The exemplary registration is a web-enabled process that requests the basic identifying information from the object owner (publisher, ad agency, studio, etc.) and returns to the registrant a packet with a unique identifier to be embedded within the object. A watermark embedding application (i.e., software) uses this packet to embed the watermark type and serial number within the client's object.

In the illustrative system, only one watermark may be embedded within a single object. In other embodiments, multiple watermarks may be embedded into a single object.

When a customer registers a watermark, the system associates the watermark serial number with the information provided by the customer during the registration process. The associated information may vary with different products. One set of associations, for the exemplary magazine advertisement objects, is shown in the following table:

Mandatory?	Information	Comments
Mandatory	Customer	Typically, the publisher
Mandatory	Publication(s)	Magazine(s) containing the ad
Mandatory	Issue Date	First date of the magazine/publication period
Optional	Volume	Magazine/publication volume information
Optional	Region Code	Optional information for regional publications
Optional	Location Code	Location of the object within the publication (e.g., page and, optionally, finer location data)
Mandatory	Watermark Type	Watermarks may have varying type. The type defines how to interpret the Serial Number
Mandatory	Serial Number	Assigned watermark number
Mandatory	Object Name	Customer's name for the object
Mandatory	Object Description	Customer's textual description of the object
Mandatory	Object Type	Ad or Editorial (and in other systems: Direct Mailer Card, Product Packaging, Coupon, Catalog, Business Card, Credit Card, etc.)
Optional	Campaign	For ads and promotions, the campaign name
Optional	Object size	Stated in fractions of the page (full page, half, etc.)
Mandatory	Effective Date	Date on which the user will first be able to initiate any actions. For publications, typically this is the "on stand" date
Mandatory	Expiration date	Date/time when the watermark expires
Mandatory	Primary Action	Initially the URL used for redirection
Mandatory	Primary Effective	Date the Primary action becomes effective
Mandatory	Primary Expires	Date the Primary action expires
Optional	Default Action	Reserved for future use (e.g., backup to the Primary action)
Optional	Default Effective	Date the Default action becomes effective
Optional	Default Expires	Date the Default action expires
Optional	E-mail address	Used to automatically notify the Customer of problems with the registration/Action
Mandatory	Status	Incomplete, active, inactive
Optional	Problem Indicator	Bad URL, slow site, etc.
Optional	Supporting Accounts	This field and its sub-fields are repeated for each supporting account
Optional	User Fields (4)	
Optional	Text	User Field free text
Optional	Viewable by Others?	Y/N. N hides the field form any other accounts

Table 1. Registration Database Elements

Watermark registrations expire -

For some products, watermarks are granted only for a limited period of time. For these watermarks, the Registration process employs an expiration date for the assigned serial number. When the system receives a message requesting action for a serial number that has expired, an error is returned. Registrants may extend their watermark serial number expirations by updating the expiration date. Expiration extensions may result in customer charges.

Watermark registration can be completed in one or more web sessions -

Registration can be a single or multi-step process. If the media owner has all of the required information at the start of the process, the system can provide a simple web-enabled method for requesting a watermark serial number (s) on-line. With all of the information provided, the registration is considered "active." That is, it is available for immediate use by the consumer. If the registrant does not have all of the required information available at the initial session, by providing a minimum set of information (e.g., name and/or organization name + product), a product watermark serial number may still be issued for the registrant to use in the embedding process. The most typical use of this partial registration occurs when the actions associated with the media to be watermarked (e.g., URL, etc.) are not yet known. The partially registered serial number is considered "inactive" until all of the required registration information has been completed. The system will issue an error message if requested to process an "inactive" serial number. Whether active or inactive, these registrations may be considered billable items subject to the terms and conditions of the applicable contract(s).

Registrations can be updated by the customer to reflect new information and/or to complete a previous registration session. For example, a registered customer may request a watermark serial number without specifying the URL used to redirect the consumer. The system will assign a serial number so that the customer can continue with the embedding process, but the registration will not be considered complete until the customer updates the registration with the URL and any other mandatory information regarding this serial number.

Watermark registrations are secure -

Only the registrant and those accounts that the registrant authorizes can access specific watermark registrations.

In the illustrated system, the customer account that registers a watermark may grant permission to a specific ad agency and/or pre-press house to change certain fields within the registration as a normal part of their work. Each customer, agency and pre-press house needs an account on file to be granted access to watermark registrations. The Customer account is established as part of the contract process. For ad agencies and pre-press houses, the accounts are established on an as-needed basis, through a controlled web site accessible to the customer.

For all products, the same basic tenet holds – access to the registration information is limited to only explicitly authorized accounts. Accounts are password protected. For ad agencies and pre-press houses, a single password may be shared. In other embodiments, each part may be assigned a unique password.

Watermark registration changes are logged –

All registration actions – creation, modification and deletion - are logged in an audit log. The authenticated username, the date/time of the action, and the action itself are all stored to provide a complete audit trail.

Processes and data flows associated with registration are illustrated in Fig. 6.

Entering Data into the Registration Database

While the client application, router, and product handler have initially been described in connection with responding to watermark information sensed from media objects, the same infrastructure can be employed earlier in the process, to enter data into the registration database 17. That is, a suitably configured variant of application 28c can be used by publishers, ad agencies, pre-press houses, etc., to (a) provide initial data to the database; (b) update such data; and (c) query the database for current values. Alternatively, a dedicated registration server 19 (Fig. 2) can be employed.

The involvement of plural parties in the registration process can be facilitated by encapsulating the database record contents for a given watermark in a file to which information is successively added (or updated) by different entities, and used to convey data between the database 17 and the cited entities.

Consider a case where Nike advertises in Wired magazine. The ad department at Wired agrees to sell space in response to a request from a media buyer at Nike. Wired may start the related watermark work by securing from the operator of system 10 a particular watermark identifier. (This, and most of the following procedures, are effected by computers talking to computers in accordance with instructions provided by suitable software used by the various participants, etc. In the discussion that follows, this software is the registration server 19 although, as noted, product handler 16 could be arranged to perform these functions.) Wired provides the operator an issue identifier (e.g., San Francisco edition of the July, 2000 issue), and internal tracking information used by the magazine. Registration server 19 responds by sending Wired a confirmatory file, by email, that encapsulates the information thus-far (i.e., the watermark identifier, the issue ID, and the magazine tracking information). Server 19 creates a new database record, and parses the received information into corresponding fields of the record.

Wired forwards the file received from the registration server to the media buyers at Nike. Nike supplements the information with its additional data, including the name of the advertisement and internal tracking information. It then forwards the updated file to server 19. Again, this server processes the file and updates the database record with the new information. It emails a confirmatory data file to both Nike and Wired, so each has the latest set of information.

The process continues in this fashion. Each entity provides new data to the registration server 19 via an emailed encapsulating file. The server updates the corresponding database record, and dispatches updated versions of the encapsulating file to the identified participants so each has the latest information.

Once Nike has entered its data via this process, it may forward the encapsulating file to its outside ad agency. The ad agency uses the file similarly, adding its particular information, and forwarding the file to the server. The server updates the database record accordingly, adds the ad agency

to its email distribution list for encapsulating files, and dispatches the latest version of the file to Wired, Nike, and the ad agency.

A pre-press house may be the next party involved, and so forth.

Identification of the URL to which the watermark ID corresponds, and updating of the database record accordingly, may not happen until near the end of the process.

At any time, any of the parties can provide additional information to the database, and share such information with others, via the same process. (Some information may not be suitable for distribution to all involved parties, and can be flagged accordingly.)

Server 19 needn't always be the hub through which all communication takes place. The file as updated by Nike, for example, can be forwarded by Nike directly to its ad agency. The ad agency can add its information, and then provide the twice-updated file to the server, etc.

By using distributed files as proxies for the actual database record, a number of advantages accrue. One is local availability of the latest information by all parties without the need for an internet connection. Thus, if a creative director wants to work on the beach, or otherwise disconnected from the net, the needed information is still available. Another is the ease of integrating software tools at each of the parties with a file of local data specific to a particular advertisement, rather than requiring the architectural hassles of interfacing with a remote database and navigating its attendant authentication and security hurdles.

While the foregoing discussion made reference to emailing files, a typical email program would not normally be used. Instead, to better manage the attendant logistics, a specialized file management/mail program is used by each of the parties. Such program would track the latest file for each advertisement, making same readily available for updating as desired, and index the files by various content fields. The user interface could thus present a list of files, grouped or sorted by any of the database fields, permitting editing or adding of information just by clicking on a given field or tab.

Of course, the file-distribution system just-described isn't essential to the system. A great variety of other arrangements can naturally be employed. One is for each party to log-on to server 19 as needed to inspect, or update, database fields for which it has appropriate permissions.

Numbering Schemes

The payload information encoded into objects (e.g., by watermarking) can take a number of forms and sizes. Four exemplary classes are discussed below:

a) Domain-based payload segmentation;

b) Customer/usage-based payload segmentation;

c) Unsegmented payload; and

d) Unique ID

Domain-Based Payload Segmentation

Domain-based payload segmentation approaches divide the payload into fields, each with a distinct meaning. The CLASS/DNS/UID arrangement earlier detailed is exemplary of this type of approach.

Consider a payload of 60 bits. Twelve bits may form a Class ID. These bits serve as an identifier for a top-level domain. 24 other bits may form a DNS ID. These bits identify an intermediate level domain. Together, the Class and DNS IDs fully identify the class of objects from which the data originates, the customer, and the server that should respond to the payload. (Some responses may be handled by the client computer, rather than dispatched to a remote server.)

The remaining 24 bits are a User ID, and serve as the most granular identifier, indicating the particular source of the payload. Based on this ID, the responding server knows exactly which response is to be provided.

This payload is embedded, in its entirety, into the customer's object. When sensed by the client computer, the application 28c parses (decodes) the payload into Class ID, DNS ID and User ID fields. The Class ID is used to trigger one or more of the client- or server-side programs. Once "launched" these products then use the Class ID in conjunction with the DNS ID and the User ID to complete the desired action.

One of the Class IDs may signify the object is a magazine page. Based thereon, the application 28c may direct the payload to the router/handler described above for response. Another Class ID may signify that the object is music. Again, the application may direct the payload to the same router. Or the application may direct the payload to a service maintained by a music industry consortium for response. Still another Class ID may signify that the object is a grocery package, and the payload should be routed to an on-line grocer for response. Yet another Class ID may signify that the object is a business card, and the payload should be processed locally, at the client machine. The mapping between Class IDs, and the corresponding response mechanism to which the application 28c should direct the payload, may be maintained by a database associated with the client computer's operating system (e.g., the Windows Registry), as detailed earlier.

Once the payload has been dispatched to a proper response destination, that entity examines the DNS ID to further classify the correct responding entity. For example, different IDs may correspond to different classes of servers within a tree of servers.

One the payload has been directed to the correct class of servers, the User ID defines the terminal "leaf" in the tree (e.g., a database record) that finally defines the response.

Customer/Usage-Based Payload Segmentation

A second approach again employs a segmented payload technique. In this arrangement, however, a first field defines the interpretation of the following bits (e.g., their segmentation into different fields).

Again, consider an exemplary payload of 60 bits. Twelve bits can be a Version ID. These bits indicate how the succeeding bits are to be parsed and interpreted, and may indicate (like the Class ID in the foregoing approach) the particular application program 28c that should be used. The Version ID bits thus serve to indicate the payload type. In the illustrated embodiment, one of these types signifies that the payload is coming from a magazine page and should be handled accordingly. In this case, the remaining 48 bits can be parsed into three fields: Owner ID (15 bits), Publication ID (15 bits), and Media ID (18 bits).

The Owner ID identifies the customer to whom the watermark is registered (e.g., Nike). This is used for ad effectiveness analysis and billing purposes. The Publication ID identifies the particular publication (e.g., July, 2000, San Francisco edition of Wired Magazine). The Media ID identifies a particular page location within that publication.

As before, the payload is embedded in its entirety into the customer's object. The payload is first parsed to determine the Version ID. If the user's device 12 has been programmed to handle such objects locally, further parsing is performed in accordance with data corresponding to that Version ID, and associated processing of the parsed data is performed. If the device has been instructed to dispatch such payloads to remote locations for service, the complete payload can be dispatched with only such further parsing (if any) as may be required to correctly identify the corresponding remote servicing entity.

Unsegmented Payload

An unsegmented payload consists only of two parts: a Version ID (as described above) and an Object ID. In an illustrative case, a 60-bit payload is again used, with 12 bits serving as the Version ID, and the remaining 48 serving as the Object ID.

In this approach the relationships of owner/customer, publication, issue, and media are all maintained in database 17 rather than literally represented in some fashion within the object identifier.

Unique ID

This case is akin to the unsegmented payload, but consists of just a single field – a unique identifier. The same application 28c is always used, and always treats the payload data consistently (e.g., processing locally, or dispatching to a predetermined destination) regardless of the payload contents.

Combinations and hybrids of the foregoing approaches can of course be used. Moreover, the 60 bit payload length is illustrative only. Longer (e.g., up to 1024 bits) or shorter (e.g., down to 8 bits) payloads can naturally be used.

In a particular embodiment, a 32-bit unsegmented payload is used, consisting of 10 bits of payload type and 22 bits of watermark serial number. Some materials (e.g., advertisements including composite graphics) may be encoded with several serial numbers. The mapping between this payload and the customer/publication/etc., is maintained in database 17.

(As noted below, the data sent from the application 28c typically includes information other than the identifier payload, e.g., the type and version number of the application 28c, the electronic address of the dispatching application, etc.)

Router

The router 14 permits any number of different products to be used by the indicia detection and response model. By keeping this function separate and generalized, new products can be added without design changes to the existing products or the product handlers 16. There are two keys to making this approach successful – speed and flexibility. By using a standardized, open interface, the router is able to facilitate both of these goals.

A premise of an exemplary interface is an enveloping technique that allows the router to "open" the outer transaction envelope and extract the vendor and application ID without decoding the remainder of the transaction (message). Given these two pieces of information, the router uses a simple lookup table to determine the product handler appropriate to complete the transaction. The router then passes the vendor, application, remainder of the transaction and the Internet "reply to" address on to the appropriate product handler. The simplicity of this handling keeps the routing delay to a minimum, while deferring the actual response processing to vendor/product-specific handlers. By including the "reply to" address in the data passed on to the product handlers, the router is freed from the responsibility of return routing for the product's response(s).

To review, the router:

1. decodes the request packet received from the client products into the packet's base components – Vendor ID, Application ID and message;

2. validates the request packet base components against a list of known, good values;

3. if a request packet component is found to be invalid, issues an error message noting the invalid components and returning same to the calling session (e.g., product);

4. sends the decoded request packet contents and any required identification of the calling session to the appropriate product handler; and

5. reports any errors encountered, including invalid packets received, to a system monitor.

Certain data flows associated with the router are shown in Fig. 5.

Product Handler

The primary function of the illustrative product handler 16 is to process requests received from the application 28c, via the Internet and router 14, and return the requested information/action to the originating device 12. In the illustrated embodiment, the information requested is the URL associated with the watermark payload sent by the application. In other embodiments, other actions and/or information may be requested.

Each received watermark payload is validated using information in the database 17. If the watermark payload ID is found and is active, the requested action is performed. If the watermark payload ID is not found or is in an inactive state, an error message is returned to the requesting application.

All requests are logged in a transaction log for tracking and billing purposes. This includes any secondary payload information (zip code, Demographic Household ID, etc.) passed in by the application 28c. The log can be maintained by the product handler 16, or elsewhere.

To speed system response, the product handler 16 may anticipatorily send URLs to the application corresponding to watermark payloads the handler foresees may be coming. These URLs can be cached in memory associated with the application 28c, and quickly recalled if needed by the application.

Consider, for example, a magazine containing watermarked advertising. If the user presents a first ad to the device 12, the watermark is decoded and forwarded to the product handler 16, which responds with a URL corresponding to that ad. The application 28c then passes that received URL to a web browser 28b on the device 12, which initiates a link to that internet address. But the handler now knows the magazine the user is reading. By reference to the watermark first received, the handler may discern, for example, that the user is reading the San Francisco edition of the March 14, 2000, Time magazine, and just looked at page 85. Based on this information the handler can query the database 17 for URLs associated with other advertising in that issue. (The database index is structured to permit fast queries identifying all ads in a given magazine issue or other collective data source.) These URLs are passed back to the application 28c and cached. If the user next presents an advertisement from page 110 to device 12, the application 28c finds it already has the corresponding URL locally cached. The application then passes the corresponding URL to the web browser. The web browser initiates the link immediately, obviating a data round trip between the application and the remote system.

The caching can be optimized in a variety of ways. One is to first send URLs corresponding to pages that are next-expected to be encountered. For example, if the user just presented page 85 to the sensor 22, after sending the URL for that page, the handler 16 would next send the URLs associated with pages 86, 87, etc. On sending the URL for the last page of the magazine (typically the rear cover), the handler could start from the beginning (typically the front cover) and send further URLs up to that for page 84. Another optimization is to first cache URLs for the most conspicuous ads, e.g., first send URLs for any 2-page spread ads, then for each full page add, then for each successively smaller fractional-page ad. Still another approach is for handler 16 to dispatch URLs to device 12 for caching in accordance with a contractually-agreed priority. One advertiser, for example, may pay a premium ad rate in exchanged for being cached before other advertisers who don't pay the premium. Other caching priorities, and combinations of such priorities, can naturally be employed.

In some systems, the advertisers or publishers may be charged for use of the system based on the number of URLs served by the system for linking. If local caching of URLs (e.g., at device 12) is employed, it is desirable for device 12 to report to router 14 (or handler 16) the URLs that are actually retrieved from the local cache and used for linking, so that the remote system can log same. Thus, each

time the user presents an object to sensor 22 for which a corresponding URL is already cached, application 28c dispatches a message to router 14 reporting the event (and, usually, the particular URL involved). This event is then logged in the transaction log.

This anticipatory dispatching of URLs is one alternative function that may be performed by a product handler. Another is if the application 28c queries the product handler to determine if a more recent version of the application is available for download. If so, the application – through interaction with the user – can request that the product handler respond with a software download.

In greater detail, application 28c can periodically query the product handler as to the identity of the latest version of application 28c (e.g., the first time the application is used each day). Device 12 may have version 3.04, and the remote system may respond that version 3.07 is the most current. In such case the application 28c can alert the user – by suitable text, graphics, or other means – that a more recent version of the program is available, and query whether such updated version should be obtained. If the user so-instructs, handler 16 can serve to device 12 the latest version of the application (or a patch permitting the presently-installed version to be updated).

Sometimes it may not be necessary to update the application version. Instead, data from the remote system may indicate the desirability, or necessity, or changing just one or more parameters in the application 28c. For example, new security keys can be dispatched periodically by handler 16 to device 12, and used to change the security configuration of the application. Or the application 28c can be instructed to direct further outgoing watermark traffic – either for the next hour, day, or until instructed otherwise - to a different router 14. Such instruction can be used to optimize system performance, e.g., for router load balancing purposes, to avoid internet routes that are found to be slow, etc.

In summary, the detailed handler:

1. validates the received identifier (e.g., watermark serial number) against the list of active identifiers; and, if the serial number is not found, return an error message to the calling session, and log the error to an error handling routine;

2. for each received, valid watermark serial number, finds the corresponding active primary action from the database;

3. for each received, valid watermark serial number, if the handler finds the corresponding primary action is currently not active, it performs an alternative, "default" action instead;

4. if the handler finds an active primary action associated with the received valid watermark serial number, it returns the URL for application use in redirection (round trip approach), or serves the HTML page found to the calling session;

5. if the handler does not find an active primary action associated with the received valid watermark serial number, but does find an associated default action, it returns that URL for application use in redirection (round trip approach), or serves the HTML page found to the calling session;

6. if the handler does not find a valid, active primary or default action associated with the watermark serial number, it returns an error message to the calling session, and logs the error to the error handling routine;

7. records each transaction, including those that result in error messages, for billing and analysis purposes (in other embodiments, this function may be performed by the router, instead);

8. responds to a "software version request" by returning the most recent available application software version number to the calling session;

9. responds to a "software download request" by initiating a file transfer of the most recent available application software to the calling session;

10. responds to a valid Request for Registration packet upload (proper format, an existing serial number, an account ID and a valid corresponding account password) by returning a current registration packet for the provided watermark serial number;

11. responds to an invalid Request for Registration packet by returning an error message to the calling session noting the failure;

12. responds to a local transaction cache flush request by writing the locally cached transactions to the transaction log; and

13. responds to a multiple URL request by returning the URL associated with the provided serial number first, followed by all other active serial numbers and URLs for the publication, issue and region code (optional) provided.

Certain of the above-described processes associated with the product handler are shown in Fig. 6.

URL Performance Monitoring

Returning to operation of the system, the URLs identified in database 17 may, from time to time, become inoperative or impaired due to equipment problems at the remote web site or otherwise. If desired, the handler 16 (or another component of the system) can be programmed to periodically test each of the links registered as active in the database (e.g., once per day), and measure the time for the associated web page to load. If the web page doesn't load, or takes much longer to load than usual (and re-tests confirm that the condition isn't an anomaly), those conditions can be flagged in the corresponding database record. If the handler is requested to provide such a URL to a device 12, the handler can send a message – either with or without the URL – indicating to the device that the URL is misbehaving.

If the URL is working, but is unduly slow to load (either compared to its historical performance, or compared to other URLs), handler 16 can provide an interim diversion to the device 12. For example, it can instruct the device to launch a second browser window, and direct that browser to an alternate destination to entertain the user while waiting for the intended page to load. When the intended page is finally loaded, the first browser window can be displayed - either by closing the second, diversionary window, or by bringing the first window to the front while keeping the second window alive in the background.

This alternate destination is desirably a low bandwidth page, so that it will not unacceptably further slow loading of the desired URL. This alternate page can be one selected by the handler, for

which the URL is sent after the desired URL. Or instead of providing a URL from the handler, the handler can serve an HTML or other page directly to the device 12. Or the alternative URL can be stored at device 12 and used to invoke the second browser window upon receipt of data from handler 16 indicating that the desired content will be slow in coming. In some embodiments the user can identify several alternate URLs (e.g., weather, stock info, jokes) and the handler or the application 28c may select among them randomly or otherwise. Or an HTML page or other application can be loaded locally at the device 12 in response to a "get ready to wait" indication from the handler 16.

If a URL is marked in the database 17 as slow or inoperative, the scanning operation periodically rechecks the URL to see if its status in the database should be changed (e.g., changed from inactive to active). Inactive URLs are reported to the registrant by email, and flagged for manual follow-up if not restored to action within a predetermined period.

Illustrative Responses by Product Handler

Part I, above, provided a sampling of the great variety of diverse applications enabled by the illustrated system 10. A few more are detailed below.

Consider use of the system 10 to enable personalized greeting cards. A greeting card company may prepare watermarked press-on stickers for use with its cards or other correspondence. The customer shows the sticker to a camera-equipped computer (either at the retail store, at home, or elsewhere). The computer decodes the watermark and sends same to a corresponding product handler 16 through the router 14. The handler – recognizing the watermark as an unregistered greeting card sticker – invites the customer to enter a destination URL, such as the customer's personal web page. This information is entered by the consumer and forwarded to the remote system for entry in the registration database 17. Thereafter, whenever the sticker is shown to a suitably-enabled system (e.g., by the card recipient), a browser window is automatically launched and directed to the web page specified by the purchasing consumer. (The same result can, of course, be effected without use of stickers, e.g., by encoding the greeting cards themselves.)

In some applications, the product handler may have a library of different responses it can provide to a user in a particular context, depending on the user's further selection. Consider a university student having a suitably-watermarked university ID card. When the card is presented to a device 12, the product handler replies with HTML instructions causing an options menu to appear on the device screen, e.g:

1. Review calendar of upcoming university academic events
2. Review calendar of upcoming university sporting events
3. Review present class schedule
4. Select courses for next semester
5. Review grades

When the student makes a selection (e.g., with a mouse, or by moving the ID card in a specified manner), the application 28c dispatches data corresponding to the selected option to the product handler, which then responds with the requested data.

In some cases (e.g., Review present class schedule, Select courses for next semester, Review grades), care must be taken to protect such information from persons attempting access using lost or stolen IDs. Accordingly, when any of these options is selected, the handler 16 may first respond to device 12 by querying for a password or PIN. Only after entry of the correct password/PIN is the requested action performed. (For security reasons, the university may prefer that the password authentication process be performed by a dedicated on-campus server, rather than by product handler 16. Naturally, this and other tasks can be delegated to processors other than handler 16 as best fits the situation.)

In other cases, an option menu needn't be presented – the correct response is inferred from the context or environment. Consider a drivers' license that is watermarked with identification of the owner. If presented to an email kiosk 12 at an airport, the decoded watermark may be used to look-up an email account corresponding to that individual, and download new mail. If the same drivers license is presented to a check-in kiosk, the decoded watermark may be used to look up that person's flight reservation and issue a seat assignment. In both cases the kiosks can be essentially identical. One, however, identifies itself to the router/product handler as an email kiosk, the other as a check-in kiosk. The response undertaken by the router/product handler differs accordingly.

Returning to the university example, there may be cases in which students are tempted to swap photos on a student ID, e.g., to permit an imposter to take a graduate school qualifying exam on behalf of a less-qualified student. In the usual case, such photo-swapping may be difficult to detect. This problem can be combated by an exam check-in procedure that includes having each student present their ID to a device 12. An application 28c specialized for this task can forward a watermark decoded from the ID photograph to a handler 16, which responds by causing an image of the identified student to be displayed on device 12. (The university could compile the requisite database of student images as it issues ID cards.) If the exam proctor sees an image on the device that does not match the image on the ID card, appropriate action may be taken. (This arrangement is applicable wherever photo ID documents are used, including airport check-in, customs, immigration, etc.)

Still another application of the illustrated system is to look-up, or act on, meta-data associated with a marked object. Consider an image, video, or audio file that a user downloads from the internet. Familiar applications such as Microsoft's Windows Explorer (including Internet Explorer) may be configured with watermark decoders activated, e.g., from a Properties panel (accessed, e.g., by right-clicking on the file icon or name and selecting the "Properties" option). When a watermark is detected in a file, the Explorer application can send a corresponding packet to a remote system (e.g., the depicted router/product handler/database). The remote system recognizes the packet as originating through the Properties panel of Windows Explorer, and looks-up the watermark ID in a database 17. Meta-data corresponding to the file (e.g., proprietor, creation date, licensing terms, exposure data, subject, etc.) is

returned from database 17 (or from another database identified by the router, handler, or database) to the application 28c, and is displayed in the Properties panel (optionally under an appropriate "tab").

(The present assignee has long offered a "MarcCentre" service that serves as a clearinghouse through which watermark identifiers found in photographs, etc., can be used to identify the proprietors and associated information corresponding to such objects.) In embodiments of the present utilizing this service, the router 14 passes the request to MarcCentre server (a product handler in this instance), which provides the solicited information back to the originating application. The present assignee's MarcSpider service complements the service provided by the Media Commerce product. The MarcSpider service constantly scans Internet sites evaluating each graphic encountered to determine whether it contains a watermark. (Audio and video can be similarly analyzed.) With each detected watermark, the MarcSpider service records the graphic file name, size, format, date/time and URL where the graphic was found. This information is then made available to MarcSpider customers in report form.)

Instead of simply displaying the meta-data, the application and/or the remote system can make use of it. For example, if the meta-data indicates that the proprietor of a watermarked image is Corbis, and that the image can be licensed for a certain use under certain terms, the remote system can be utilized as a licensing server – receiving payment information from the user, granting the license, and forwarding transaction details to Corbis.

Still another application is the sale or promotion of music or video over the internet. Taking the case of music, an artist may freely distribute a low-fidelity (or otherwise corrupted or abridged) version of a song. The low fidelity can be by reason of bandwidth limitation (e.g., 500Hz – 2.5 KHz), monophonic (as opposed to stereo), or otherwise. The artist can seek to distribute the low-fidelity version as widely as possible, to serve as a marketing agent for the artist's other works. (The free distribution of lower-bandwidth audio may serve to alleviate some of the network bandwidth problems faced by universities whose students actively engage in transferring free music over the internet.)

Each low-fidelity version can be processed to extract an identifier (e.g., a steganographic in-band watermark; a numeric ID or song/artist name field in a in a file header; a 128-bit hash value obtained by applying a hashing algorithm to the music data, the music file header data, a portion thereof, etc.) If a listener is interested in obtaining a full-fidelity version of the work, the listener can operate a suitably programmed computer or music appliance that extracts the identifier from the work and passes it on to the remote system. The remote system can respond in various ways, e.g., by providing a full-fidelity version of the same work back to the user (such as MP3 download) and charge the user's credit card a fee (e.g., $0.99); or by directing a web browser on the user's computer to an e-commerce/fan web site associated with the music, etc. Such functionality can be provided in general purpose programs such as Microsoft's Internet Explorer, e.g., by right-clicking on a file to obtain a menu that includes this and related functions.

Figs. 8-10 show a sequence of screen shots from such an embodiment. In Fig. 8, a user has right-clicked on an MP3 file icon in a directory listing 200. A property menu 202 pops up that includes, as its second option "MP3Bridge."

Fig. 9 shows what happens when the user selects the MP3Bridge option. An MP3 player 204 is launched, and a dialog box 206 appears. The dialog box queries the user, "More Information About the Artist? Yes No."

Fig. 10 shows what happens if the user selects "Yes." The software sends the identifier – extracted from the MP3 file – to the remote system. The remote system responds with the address of an associated web page, and instructs the user's computer to launch a new browser window directed to that page.

The same functionality can naturally be invoked through the user interface of the MP3 player (or a Windows MediaPlayer, etc., rather than through Internet Explorer). The music application can spawn a separate window, or present the options and the associated data within the existing window.

Yet another application of the remote system is as a "net nanny" filter. Links requested through the system can be checked for keywords, adult-content flags, content ratings, or other indicia of age-suitability, and provided to the requesting computer 10 only if they meet certain earlier selected criteria.

Again, it will be appreciated that the foregoing examples are but a few of myriad applications enabled by the detailed system.

Reporting

System software may enable the provision of customer-accessible reports (accessible over the internet) that show detailed and summary usage information by date, customer, publication, issue date, region, product/version, etc. These reports can be both regularly scheduled and ad-hoc. The specification of the content, relationships and the timing of the reports can be defined by the customer on-line.

Illustrative reports detail:

a) Hit rates/Transactions per customer per ad
b) Hit rates/Transactions per customer per publication per ad
c) Hit rates/Transactions per customer per publication per issue per ad
d) Hit rates/Transactions per customer per publication per issue per region per ad
e) Hit rates/Transactions rates by originating application (28c)
f) Hit rates/Transactions by originating application vendor
g) Hit rates/Transactions rates by originating web domain (e.g., aol.com)
h) Hit rates/Transactions rates by postal/zip codes
i) Hit rates/Transactions by country

Additional marketing/marketplace reporting can also be produced for internal analysis by the service provider, and for sale to other entities. These reports typically provide a more global view of the impact and usage of the system. Using information stored in a demographic database, in conjunction with these usage patterns, the system can provide customers and research agencies with more detailed demographic/statistical data on the system's usage and effectiveness.

In an illustrative system, certain statistics in the demographic database are compiled using statistics from a sample of users that consent to have their activities tracked in some detail, in

consideration for certain perks (e.g., give-away cameras, bar-code scanning pens, or other devices, etc.). These users are termed Demographic Households. A software program included in the systems solicits information detailed in the following table from such users over the internet, with a web-enabled interface. A related program allows such users to update/edit their user/household information previously entered. Each such session is password authenticated for security.

User Information	Comments
Name	
Address	
Street	
City	
State	
Country	
Postal Code	
Phone number	
E-mail address	
Household Annual Income	Provided as raw $ or as a selection from a range of numbers
Occupation	
Education	May be per member of household.
Profession	If applicable
Number of members of household	
Member of household	
Age	
Sex	
Internet user?	
User of this linking service?	
Internet usage per week	In hours. Sum of entire household
Internet business usage per week	
Primary Internet usage?	Typical household use of the Internet. May be a selection list
Owned a computer since?	Year only
Number of computers in the home?	
Types of computers in the home?	Mac, PC, etc. Select all that apply
Rooms where the computers are located	Home office, bedroom, etc. Select all that apply
What ISP do you use?	
What is your modem speed?	Select from list that includes ISDN, ADSL, cable + dial up modems.
Are you willing to be an official "Demographic Household" and allow us to contact you for feedback on our products and advice on new products?	
What other technology devices do you have in your household?	Scanners, PC cameras, digital cameras, DVD, PDAs, etc. Select all that apply

Audio and Video

As with paper advertisements, the illustrated system provides users of web-connected PCs or other appliances with methods of obtaining information, content, associated products, or associated services using the same principles as detailed above.

For example, an application 28 can "capture" music or other audio using a recording device (note recorder, microphone connected to PC, MP3 player, etc.) and analyze the captured audio to detect an embedded watermark. Once detected, the application passes some or all of the watermark payload information, together with identification of the application and its vendor, to the router. The router forwards the payload information to a handler corresponding to the application. The response of the product handler varies with the context and nature of the data. For example, the handler may return the artist, title, track, album, web URL and purchasing information, to the user. Recorded news and entertainment segments may include transcripts (audio, video and/or text) of the segment along with other related web site information. The handler may simply cause the device 12 to launch a browser window directed to a music commerce web site where the music can be purchased.

Security

The system's basic security philosophy is to grant access to each customer's information only to the users authorized by the customer. To this end, the system desirably should:

1. Create and maintain a list of authorized users (accounts).
2. Employ security methods to deny access to any unauthorized users.
3. Limit users to access only the objects they are authorized to access (typically, the objects belonging to that customer).
4. Report and record all unauthorized access attempts.
5. Maintain a log of all authorized user logins (sessions).
6. Provide the capability for the watermark registrant to grant access rights to other accounts (such as ad agencies and pre-press houses).
7. Establish initial passwords for each account
8. Provide the capability for each authenticated user/account to change their password
9. Provide the capability to reset an authenticated user/account's password in the event the current password is lost.
10. Store all passwords as encrypted values (to prevent theft of passwords).
11. Provide the capability to restrict the creation, modification, deletion, and listing/viewing of account information to authorized users.

Audit Trail

Because of the financial implications of the system's activities, all changes to any registration or customer data need to be recorded. This audit trail provides the operator and its customers with an accurate accounting for the current and previous states of the data.

The audit software desirably records the creation, modification, and deletion of all registration and customer data. The audit software also records the username, date/time of creation/modification/deletion of records, and – for modifications – the before and after images of the data changed.

Application-to-Product Handler Interface Definition

The basics of the interface between the application 28c and the handler 16 are (a) a flexible request and response package structure, and (b) a defined connection method based on industry standards. The illustrated messaging employs the http and/or the https protocol to send and receive messages among the system components. An overview is provided in Fig. 4.

Message Format

The message format is XML-compliant and is defined by the following XML DTD –

```
<!DOCTYPE list [
<!ELEMENT Content (vendor, appl, prod)>
<!ELEMENT vendor (#PCDATA)>
<!ELEMENT appl (#PCDATA)>
<!ELEMENT prod (#PCDATA)>
]>
```

The application 28c appends its data to this header for transmission to the product handler 16. Exemplary messages and product handler responses are detailed in the sections that follow.

Application Message Definitions

The application message definitions can be broken down into Request Code, Primary and Secondary information.

The **Request Code** instructs the product handler 16 to take a specified action.

The **Primary information** portion contains the data required to properly service the application's request. The Primary Information varies based on the Request Code.

The **Secondary Information** is intended for use by analysis and reporting tools and does not instruct nor aid the product handler in servicing the user's request. Secondary Information contents change based on the Request Code and not all Request Codes are required to have associated Secondary Information. In addition, most of the Secondary Information requires the consumer to grant express consent to its collection. If that consent is not given, the application does not send Secondary Information. A special case exists for selected, consenting consumers to become part of a demographic database.

Primary and Secondary information may change by request type, but in general conform to the definitions below. The generic format for the product handler is also defined below.

Primary Information includes the Application Version, Watermark Type, Watermark Serial Number, Context and Environment.

- Application Version: used by the product handler to modify its actions, typically for backwards compatibility
- Watermark Type: top 9 bits of the illustrative watermark payload. Used by the product handler in processing the Watermark Serial Number
- Watermark Serial Number: remainder of the watermark payload. Provides the index used by the product handler to access the watermark in the registration database
- Context: instructs the product handler to modify/refine its action based on the consumer request's context
- Environment: instructs the product handler to modify/refine its action based on the consumer request's environment. (The environment may be specified, e.g., as home, office, car, portable appliance, etc.)

Other Request codes can, of course, be used. Each may have its own list of mandatory and optional Primary Information fields. Optional fields are excluded from the primary Information when there is no value associated.

Secondary Information:

- Demographic Household ID: identifier for a selected demographic group. This is used as an index to the actual demographic
- Input device: Manufacturer, model and version of the device used to detect the watermark (e.g., a TWAIN driver string)
- Operating System: operating system in use on the consumer PC
- Processor: processor type/class on the consumer PC
- Processor speed: processor clock speed, in MHz, of the consumer PC. (May be entered by user, or auto-detected.)
- Language: preferred consumer spoken language
- Country: Country where the consumer PC resides
- Postal Code: Consumer's postal code (used along with the country to pinpoint the location of the consumer).

(In addition to these explicit data, the packet sent from the device 12 also conveys an IP address (inherent in the use of http protocols) so that the remote device (e.g., the router/handler) has an address to which it can respond.)

Response from Product Handler

```
RtnCode          - Success =1
```

URL	- the active URL for the watermark serial number received
	or
RtnCode	- Error <0
Error Message	- text.

Request for URL

Required Inputs

Header (XML format)

Vendor	(e.g., = Digimarc)
Appl	(e.g., = MB)

Data

Required information –

Req	=RFU
Ver	= application version number
Type	= watermark type number
Ser	= watermark serial number
Cxt	= context
Env	= environment

Optional Information –

Ctry	=User's Country name
Lang	=User's preferred Language
HHID	=Demographic Household Identifier
Det	= TWAIN string of the sensing/detecting device
OS	=User PC Operating System string
Proc	=User PC processor type and class
Speed	=User processor speed
Zip	=User postal code

Example:

```
<?xml version="1.0"?>
<Content>
<vendor>Digimarc</vendor>
<appl>MB</appl>
</Content>
Req=RFU
Type=1
Ser=10001
Ver=1.0
Cxt=A
Env=Q
Ctry=USA
Lang=English
HHID=1234567
Det=TWAIN string
OS=Win98
Proc=Pentium III
Speed=500
zip=74008-1234
```

Response from Product Handler

RtnCode=Success/Error number (Success = 1)
URL=*URL associated with specified watermark type and Serial number*
Exp=*Expiration date/time (GMT) for caching purposes - format of mm/dd/yyyy hh:mm:ss*

or

RtnCode=Success/Error number (Error <0)
MsgText=message text

Error reasons:

-1 Type and Serial Number OK, but no URL in database (both the primary and default URL are missing)

-2 Type and Serial Number OK, but URL is marked as inactive (neither the primary nor the default is active)

-3 No record in database matching the Type and Serial Number

-4 Request format error – incomplete data

Request for Configuration

Required Inputs

Header (XML format)

```
Vendor      (e.g., = Digimarc)
Appl        (e.g., = MB)
```

Data

Required information –

```
Req         =RFC
OS          =User PC Operating System
```

Example:

```
<?xml version="1.0"?>
<Content>
<vendor>Digimarc</vendor>
<appl>MB</appl>
</Content>
Req=RFC
OS=Win98
```

Response from product handler

RtnCode= Success/Error number (Success = 1)
Ver=*Latest Application version# available for download*
https=yes (or n)
GCURL=URL used to route subsequent Application requests

or

RtnCode= Success/Error number (Error <0)
MsgText=message text

Error reasons:

-5 Unknown Operating System

-4 Request format error – incomplete data

Request for Associated URLs

Required Inputs

Header (XML format)

```
Vendor      = Digimarc
Appl        = MB
```

Data

Required information –

```
Req         =RFA
Ver         =application version number
Type        =watermark type number
Ser         =watermark serial number
Cxt         =context
Env         =environment
```

Example:

```
<?xml version="1.0"?>
<Content>
<vendor>Digimarc</vendor>
<appl>MB</appl>
</Content>
Req=RFA
Type=1
Ser=10001
Ver=1.0
```

Response from product handler

```
RtnCode= Success/Error number (Success = 1)
Ser1=watermark serial number
Type1=watermark type number
URL1= URL associated with specified watermark type and Serial
number
Exp1=Expiration date/time (GMT)
Ser2=watermark serial number
Type2=watermark type number
URL2= URL associated with specified watermark type and Serial
number
Exp2=Expiration date/time (GMT)
…..
Ser'n'=watermark serial number
Type'n'=watermark type number
URL'n'= URL associated with specified watermark type and Serial
number
Exp'n'=Expiration date/time (GMT)
      or
RtnCode=Success/Error number (Error <0)
MsgText=message text
```

Error reasons:

-8 Type and Serial Number OK, but no associated watermarks or URLs in database

-9 Type and Serial Number OK, but all associated URLs are marked as inactive

-3 No record in database matching the Type and Serial Number

-4 Request format error – incomplete data

Request for Transaction Download

(Needed to account for locally cached redirections. One request per local redirection.)

Required Inputs

Header (XML format)

```
Vendor      = Digimarc
Appl        = MB
```

Data

Required information –

```
Req         =RFT
Ver         =application version number
Type        =watermark type number
Ser         =watermark serial number
Cxt         =context
Env         =environment
```

Optional Information

```
Ctry        =User's Country name
Lang        =User's preferred Language
HHID        =Demographic Household Identifier
Det         =TWAIN string of the sensor device
OS          =User PC Operating System string
Proc        =User PC processor type and class
Speed       =User processor speed
Zip         =User postal code
```

Example:

```
<?xml version="1.0"?>
<Content>
<vendor>Digimarc</vendor>
<appl>MB</appl>
</Content>
Req=RFT
Type=1
Ser=10001
Ver=1.0
Cxt=A
Env=Q
Ctry=USA
Lang=English
```

```
HHID=1234567
Det=TWAIN string
OS=Win98
Proc=Pentium III
Speed=500
zip=74008-1234
```

Response from product handler

```
RtnCode=Success/Error number (Success = 1)
            Or
RtnCode=Success/Error number (Error <0)
MsgText=message text
```

Error reasons:

-4 Request format error – incomplete data

To provide the fastest possible system response, it is desirable that data exchanges between the originating device 12 and the remote system be as short as possible – preferably of a size that can be transported in a single internet data packet (i.e., less than about 536 bits). Such an arrangement avoids the overhead associated with data division on transmission, and data reassembly on reception.

Generally speaking, the combined elapsed time of the system service (i.e., watermark recognition by application 28c, packet delivery to router, decoding by router, handling by product handler, and return of response to application) for a single request should average no more than 3 seconds as measured from receipt of request to 1st byte sent in response to request. Typical speeds are less than 2 seconds, with many responses being provided in less than 1 second.

The immediately following discussion reviews much of the foregoing, but from a different priority case and with additional details.

The MediaBridge digital message is composed of two codes, both of which are embedded in MediaBridge-enhanced images. The media *owner code* is assigned by the system administrator (e.g., Digimarc) and identifies the entity licensed to add MediaBridge enhancement to images. The *routing code* is assigned by the media owner (advertisers, publishers, manufacturers, etc.) and determines where the end-user will be directed when the Client Application reads the MediaBridge codes. Two different advertisements embedded with the same codes can take the end-user to the same web page, while the same advertisement in different publications using different codes can go to different web pages and can be used to track which advertisement or magazine is most effective in bringing end-users to the media owner's web site.

There are three major components of the MediaBridge system. The Client Application is used in homes and businesses by consumers (the MediaBridge end-users) to automatically navigate from an image or object to additional information, usually on the Internet. The MediaBridge Router provides the Client Application with the appropriate Internet address for a given image or object. The Embedding System is used by the media owners to embed the MediaBridge codes into images prior to printing.

MediaBridge Client Application

The Client Application can be distributed via OEM relationships with tethered digital camera manufacturers such as Logitech, 3Com, Creative Labs and others. It is installed by the end-user along with the camera driver and supporting software from the camera manufacturer's installation CD.

The MediaBridge Client Application may run on 200 MHz or faster Pentium or PowerPC computers configured with a tethered video camera under the Windows 95/98/NT 4.0 and Macintosh OS 8.6 operating systems.

The MediaBridge Client Application is initially focused on using MediaBridge-enhanced images to directly browse to additional information on Internet web sites. Therefore, it requires a connection to the Internet, either through a dial-up modem or a permanent connection. However, the Client Application desirably has an extensible architecture that also supports browsing to local data, e.g. on a CD, and linking to other applications.

The MediaBridge system can be used almost anywhere that a video camera can show clear, focused, high-quality images. It operates under lighting conditions varying from dim (about 40 lux) to bright (3900 lux). In very dim light it works with the aid of a supplemental light source, and in bright glare it works if the pictures or objects are shielded from the light source.

Operating the End-user Client Application

The first time the Client Application runs it desirably presents the user with a wizard that teaches the best techniques for presenting MediaBridge-enabled images to the Client Application through the video camera. The wizard is tuned for each camera and can use either the sample images or live production images.

Shipped on the installation disk (e.g., CD) can be one or more games that teach users how to best use the Client Application – e.g., focusing the camera, positioning the encoded object in the camera's focal region, holding the object stationary for the second or so needed for decoding, permissible lighting requirements, etc. The camera box, or inserts inside the box, can be MediaBridge encoded and can link through the browser to a corresponding introductory page hosted by the camera vendor or the system administrator. T-shirts or other prizes may be awarded through the web site, either on a random basis, or upon the showing of proficiency in some aspect of MediaBridge operation.

In an illustrative configuration, the MediaBridge Client Application is always running and is either active or inactive. When active, the MediaBridge video camera window is always on top of any other windows and fully visible, and the Client Application is constantly checking the video for a MediaBridge-enhanced image. When a MediaBridge-enhanced image is found, the appropriate information is displayed - most often as a web page - and the Client Application optionally becomes inactive. When the Client Application is inactive, it hides the video window, releases the camera so it is available for other applications, and uses very little memory or computer resources.

In one implementation, the Client treats the camera as a serially re-usable device. That is, when the Client Application is active and checking the video, no other application can access the video camera. Similarly, the Client Application cannot access the video when another application is using the camera.

The MediaBridge Client Application may include the following functionality:

1) **Browser/Application Launch**. When a MediaBridge-enhanced image is found, the Client launches the user's web browser or another application. When browsing to an Internet web site, the Client provides the site with the identity of the image and the end-user's zip code, if available. This information allows the media owner's web site to display a localized web page to the user. When launching another application, the Client provides the image owner code and the routing code to the application.

2) **Destination menu**. If multiple URLs are specified for a routing code, the Client Application displays a browser-based menu that allows the end-user to chose what page to display, rather than directly navigating to a web page.

3) **Branding**. While retrieving the information for displaying a web page, the Client Application displays a local web page that with a pre-stored brand (e.g., Digimarc) and explains that the desired information is being retrieved. This branding page is replaced with the media owner's page without causing any delay to the user. (The contents of the branding page may be updated during Automatic Software Update.)

4) **Flexible activation**. When the Client Application is inactive, the user can activate it by:
   a) Clicking on the application icon.
   b) Pressing a hot-key combination.
   c) Clicking on a tray icon. (Windows.)
   d) Clicking a button on the browser toolbar.
   e) Restoring the minimized MediaBridge Client video camera window. (Windows.)

5) **Status display**. While active, the Client Application provides feedback to the user through the status display pane, which is displayed in the video camera window, next to the view as seen from the camera. The status includes feedback on the ambient lighting conditions, the distance from the camera (if available,) focus, and the current state (preparing to display the image-related information, trying to read the image, or waiting for a MediaBridge-enhanced image to be presented to the camera.)

6) **Multiple camera support**. An illustrative MediaBridge Client Application uses one camera at a time, but can use any MediaBridge-enabled camera installed. If a computer only has one camera, that camera is automatically selected. If more than one camera, the user can select which camera will be used and can change the selection while the Client is running.

7) **Automatic Software Update**. The user can automatically install updates to the Client Application by making a simple menu choice. The Client connects to the Internet, and downloads and installs any available updates. In addition, the Client will suggest checking for updates if it finds a

MediaBridge-enhanced image that uses a newer protocol (e.g., by reference to a version indicia encoded with the MediaBridge data).

8) **User option configuration.** Nearly everything in the Client Application can be configured by the user.
   a) Automatically starting the Client whenever the computer starts. (Default.)
   b) Activation by hot key.
   c) The display of an activation icon on the system tray. (Windows.)
   d) Adding a button to the Internet web browser. (Internet Explorer 4 or later, and Netscape 4 or later.)
   e) Selection of the last camera used on startup when multiple cameras are installed.
   f) Reminder to check for software updates periodically.
   g) Running the Wizard at startup. (Default)
   h) Automatic deactivation when a MediaBridge-enhanced image is read.
   i) Automatic deactivation if nothing is presented to the camera for a specified period.
   j) Beep or play sound file upon reading a MediaBridge-enhanced image.
   k) Blocking of sites based upon RSACi ratings. Each media owner is expected to self-rate its site using the RSACi scale for the categories of language, nudity, sex and violence. The Router blocks the end-user from a site if the RSACi rating for any category exceeds what the end-user allows. (The Router relies solely on the media owners' self-ratings and does not actually check web pages for RSACi codes.) Selecting this option also results in the display of information about RSACi (http://www.rsac.org/ratingsv01.html) regarding restricting content within the web browser.
   l) Automatic connection to the Internet over an existing dial-up (modem) connection.
   m) Password protection upon startup.
   n) User registration information. (The user is informed that all information is optional and that specific items will be provided to third parties if provided.)
   o) The end-user zip code, which the user is not required to provide, is used as a basic piece of demographic information passed along to media owners if it is voluntarily provided by the end-user.

9) **Wizard.** This wizard provides guidance in setting up a particular camera to obtain the best results with MediaBridge and the best techniques for using MediaBridge as a portal to more information. By default the wizard runs each time the Client Application is started. It also runs the first time a new camera is selected by the user.

10) **Camera verification.** When a new camera is selected as the MediaBridge Client input, the Client verifies that the camera is MediaBridge-enabled. If it is not, the user is warned that the camera is not supported and that it may not work correctly with the Client Application.

11) **Extensible architecture.** The MediaBridge Client handles reading MediaBridge-enhanced images and connecting to the desired information, usually by displaying Internet or local web pages in a

browser window. Its functionality can be extended by MediaBridge-enabled applications that register for the handling of specific data. For example, if a MediaBridge-enhanced business card were presented, the MediaBridge Client could run a MediaBridge-enabled application that downloads business card information from a web site and updates the user's contact list with the new information. Another example would be a page from a MediaBridge-enhanced children's book that causes an audio file to be played for the page.

12) **Embedding support**. If the Client Application is being used by a media owner to verify that an image has had the correct MediaBridge codes embedded in it, the Client displays the name of the media owner, the routing information and the relative strength of the MediaBridge watermark when a MediaBridge image is read. In order to protect the privacy of each media owner, this information is only provided if the user is able to provide a valid embedding user name and password for the media owner.

MediaBridge Router

The Router is essentially transparent to the end-user. When the Client Application detects a MediaBridge-enhanced image linked to information on the Internet, it communicates with the Router to obtain the Internet addressing information of the web page to be displayed.

The Router also includes the following functionality:

1) **Routing information maintenance**. Contained within the Router is information linking each unique MediaBridge code to one or more related Internet addresses. In one implementation, the system administrator (e.g., Digimarc) maintains the routing information using information supplied by its advertising customers. In another implementation, the media owners update the information themselves using a secure Internet connection with the Router.

2) **Problem handling**. If a routing request cannot be satisfied, the Router responds in a way that minimizes the impact on the end-user. For example if the MediaBridge code is unknown, then the Client Application is given the URL for the media owner's home page. In addition, each day the error occurs the Router notifies the media owner by email that the error occurred.

3) **Content rating**. If the media owner has provided RSACi rating information for the image or its site, and the end-user has specified RSACi ratings that specify that the site is to be blocked, the Router returns a web page to the user indicating that the site contains inappropriate content and does not connect to the media owner site.

4) **Validation**. Periodically (e.g., on a daily basis) the Router validates all of the active information in its database. If any errors are found, the media owners are notified by email. If the errors are not fixed with one day, Digimarc is notified. The conditions checked are:
   a) Missing URLs on active links.
   b) URLs referring to non-existing pages.
   c) Pages that exceed the acceptable download time as required by Digimarc.

5) **Tracking**. The Router may log Client Application requests in order to develop marketing information. Such a tracking log could include:
   a) Date and time of the request.
   b) Media owner and image.
   c) Zip code of the end-user, if provided.
   d) IP address that issued the request.
6) **Reporting**. Using the tracking information, the Router provides the following reports for use by Digimarc:
   a) Number of URL requests for a particular date range.
   b) Number of URL requests by media owner for a particular date range.
   c) Number of URL requests by MediaBridge-enhanced image for a particular date range.

(The tracking and reporting are desirably structured to allow media owners to obtain traffic and marketing reports online from the Router.)

MediaBridge Embedding System

The Embedding System includes:

- a Photoshop-compatible Embedder Plug-in for embedding the MediaBridge codes,
- a Photoshop-compatible Reader Plug-in for verifying MediaBridge-enhanced digital images,
- the Client Application for verifying MediaBridge-enhanced proof and press prints, and
- an Internet-based Router Maintenance Application for obtaining MediaBridge codes and assigning URLs to MediaBridge codes.

The Plug-ins require a connection to the Internet only for authorization and to assign new MediaBridge routing codes. The Router Maintenance Application always requires an Internet connection.

Router Maintenance Application

The Router Maintenance Application's primary purpose is to assign routing information to the MediaBridge codes so that an end-user is presented with the appropriate web page when the Client Application reads an image.

The Router Maintenance Application includes the following functionality:

1) **Multiple locations**. A single media owner account can be accessed by different people in different locations. It can also be accessed by people in different organizations such as by an advertiser and its advertising agency.
2) **Restricted Access**. Each media owner has the ability to restrict who can access the Router information and can specify who can create new MediaBridge codes, use existing MediaBridge codes, or change the information for routing codes. The media owner can add, change or revoke access at any time.

3) **Secure access.** All Internet access to the Router is through secure connections.

4) **Routing Codes.** The assignment of new routing codes, changing existing routing and deleting or reusing old routing codes.

5) **Time-based routing.** Each routing code can be assigned multiple URLs, each of which can optionally have an effective date and an expiration date that determines when it is to be used in routing. Expired URLs may automatically be deleted by the router after thirty days.

6) **Multiple routing.** If multiple URLs are specified for a routing code, the Router returns to the end-user a browser-based menu that allows the end-user to chose what page to display, rather than directly navigating to a web page. In one embodiment, a single routing code may have up to four unexpired URLs. Each URL can have a short description that will be a browser link to the URL, a long description of up to 500 characters, and a URL to an icon to be displayed in the menu. The icon can be no larger than 50 pixels high by 300 pixels wide.

7) **Logging.** All changes to the routing information are logged and may be reviewed by the media owner.

Plug-in Overview

The first time embedding is done on a computer for a media owner, the Embedder requires the account number for the media owner, and a valid user name and password as defined by the media owner. The Embedder connects via the Internet to the Router, verifies that the user is authorized, and downloads information about the media owner. Further embedding for that media owner can be performed without connecting to the Router. However, each time the user connects to the Router to obtain routing information for embedding, the Router validates the user name and password against the currently valid names and passwords. If the previously used name and password are no longer valid, the user has to provide a valid name and password in order to continue embedding for the media owner.

The media owner account number is only needed for the initial authorization since any future verification is done using the name of the media owner as maintained by the Embedder. Since a graphics artist at an advertising agency can work on projects for two or more media owners, a single graphics artist may have different user names and passwords for each media owner.

The Reader Plug-in is used primarily for verification of images after embedding. It restricts access in the same way as the Embedder Plug-in with the additional restriction that it will provide information about an image only if the user can provide a valid user name and password for the media owner. If the media owner information is not available on the current computer, the Reader will connect to the Router for verification.

The plug-ins share this functionality:

1) **Multiple media owners.** A user can perform embedding or reading for multiple media owners. The user must have a valid user name and password for each media owner.

2) **Multiple users.** A single computer can be used by several people. Unless they are using the same user name and password, they do not share information.

3) **Automatic Software Update**. The user can automatically install updates to the plug-ins by making a simple menu choice. The plug-ins connect to the Internet, and download and install any available updates.

4) **Secure access**. All Internet access is through secure connections.

Embedder Plug-in

The Embedder Plug-in also includes the following functionality:

1. **Routing codes assignment**. The user selects a routing code to embed from a list of existing codes provided by the Router via an Internet connection. Each routing code is identified by a unique routing number and description. Once selected through the Router the information about a routing code is retained on the local computer until removed by the user.
2. **New routing codes**. If authorized, the user can connect to the Router through the Embedder to create a new routing code. The routing code can be created with or without the URL information, which can be added later.
3. **Routing code updates**. Whenever the Embedder connects to the Router, it downloads any updates that were made to routing codes being cached locally.
4. **Masked embedding**. The user can mask off a part of the image so that the MediaBridge codes are only embedded in the masked area. In one embodiment, different codes are applied in different mask areas. Typically, the user cannot embed different MediaBridge codes in the same part of an image.
5. **Variable intensity**. The user can globally vary the intensity (and thus visibility) of the MediaBridge watermark from light to very heavy, and can also locally vary the intensity in different areas of an image (i.e., the intensity is adapted to local image characteristics).
6. **Logging**. Each time the user embeds in an image, the Embedder records a log of the date, time, information embedded, the embedding settings (e.g., intensity), the user name, the computer name, and the name of the input image file. The log is a text file on each user's computer. It can be viewed in any text editor.

Reader Plug-in

The Reader Plug-in also includes the following functionality:

1) **Reading**. Reads a digital image, which may have been scanned, and, if the user is authorized, displays the media owner, routing information and a measure of the strength of the MediaBridge watermark.

2) **Masked reading**. The user can mask off part of the image and read the MediaBridge information only from that part.

PART III

Inventive Combinations

In addition to the inventive aspects detailed earlier, some of the inventive combinations claimed in the priority cases include the following:

A1. A method comprising:

presenting a business card of an individual to an optical sensor, the optical sensor producing output data;

decoding steganographically-encoded plural-bit data from the sensor output data; and

using said plural-bit data to establish a link to an internet address having data relating to the proprietor of said business card.

A2. The method of claim A1 which includes obtaining from said internet site calendar data detailing certain activities of the individual.

A3. The method of claim A2 in which the amount of calendar data obtained depends on an authorization level.

A4. The method of claim A3 in which the authorization level is reflected in the plural-bit data encoded in the individual's business card, wherein an individual can distribute differently-encoded cards to different recipients, to grant the recipients different access rights to said calendar data.

A5. The method of claim A1 in which the optical sensor is a business card reader that also serves to input textual information from business cards into a personal information manager.

A6. The method of claim A5 which includes storing an internet address discerned from the steganographically-encoded plural-bit data into said personal information manager.

A7. The method of claim A1 in which the optical sensor is a digital camera.

A8. The method of claim A7 in which the digital camera is mounted to a computer display device.

A9. The method of claim A7 in which the digital camera is head-mounted on a user thereof.

A10. A method comprising:

presenting a printed promotion to an optical sensor at a first site, the optical sensor producing output data;

decoding steganographically-encoded plural-bit data from the sensor output data;

using said data to establish a link to an internet site relating to a company, product, or service promoted by said printed promotion; and

transferring to from the internet site to the first site additional information relating to said company, product, or service.

A11. A method comprising:

presenting a printed identification badge to an optical sensor, the optical sensor producing output data corresponding to optical characteristics of the face of said badge;

decoding steganographically-encoded plural-bit data from the sensor output data;

checking the plural-bit data to determine whether it corresponds to a valid access card; and

unlocking a lock depending on the outcome of the foregoing checking operation.

B1. A method comprising:

presenting a digitally encoded object to an optical sensor, the optical sensor producing output data;

decoding plural-bit data from the sensor output data; and

using said plural-bit data to establish a link to an internet address having data relating to said object.

B2. The method of claim B**Error! Reference source not found.** in which the object is steganographically encoded with said plural-bit data.

B5. An apparatus comprising

a refrigerator;

an optical sensor associated with the refrigerator, the sensor being adapted to provide image data corresponding to an object held in front of the sensor; and

an image processor for processing the image data and extracting plural-bit binary data therein;

wherein the apparatus is useful for compiling a listing of groceries.

B6. In a method of operating a computer, the computer including an operating system with a registry database, the registry database associating specific data types with specific software programs particularly corresponding thereto, an improvement comprising:

providing a frame of image data;

decoding plural-bit identifier data from the image data;

consulting the registry database to identify a software program corresponding to said identifier data; and

invoking the identified software program.

B7. The method of claim B6 which includes:

steganographically decoding plural fields of data from the image data, one of said fields comprising the identifier data; and

providing another of said fields of data to the identified software program for use thereby.

B8. A system for responding to an object encoded to convey plural-bit auxiliary data, the system including:

an optical capture device that produces image data corresponding to a field of view, the field of view including visual clutter as well as the object;

a visual clue-based subsystem for identifying the object from the visual clutter in the image data; and

a processor for extracting the plural-bit auxiliary data from the object as identified by said subsystem and for applying the extracted auxiliary data to a response system.

B9. The system of claim B8 in which the visual clue-based subsystem identifies the object from the visual clutter by reference to a spectral attribute.

B10. The system of claim B8 in which the object includes a retro-reflective member, and the system includes a source of illumination of spectral character that is reflected off the retro-reflective member and detected by the optical capture device.

B11. The system of claim B8 in which the visual clue-based subsystem identifies the object from the visual clutter by reference to a temporal attribute.

B14. A method of producing an object customized to authorize a person's entry to a first location, comprising:

providing the person an access code;

generating printable data in which the access code is steganographically encoded; and

using a printer at a second location remote from the first location to print said printable data.

B15. The method of claim B14 in which the access code is provided to the person via a networked computer, and the first location is a theater.

B16. A printer responsive to commands from printer driver software, the printer driver software being characterized by being responsive to two types of input data, one of said types comprising text data, the other of said types comprising watermark data, the two types being separately applied to the printer driver software, rather than integrated together prior to application to the printer driver software.

B17. In a method of producing printed output using a computer, the printed output comprising text, the computer including software, the software presenting a print dialog box in response to user control, an improvement wherein the print dialog box has a graphical user interface permitting the user to select steganographic encoding of the printed output.

B18. A method of printing magazines, comprising:

processing blank pages to steganographically encode plural bit data thereon;

thereafter, printing text and/or imagery on the processed pages to yield steganographically-encoded printed magazine pages; and

binding at least one of said steganographically-encoded printed magazine pages into each of plural magazines.

B19. A method of travel promotion, comprising:

steganographically encoding a travel photograph to hide plural-bit data therein;

processing the travel photograph to extract the plural-bit data therefrom;

using at least part of the extracted plural-bit data to direct an internet web browser to a web site that provides travel information useful to a consumer who wishes to visit the location depicted in the photograph.

B21. A gestural method for providing input to a system, comprising:

holding an object in the hand of a user, positioning the object to be at least partially within the field of view of an optical scanning device, and manually moving said object;

with the scanning device, grabbing frames of image data plural times per second;

processing the frames of image data to identify the object as represented therein;

analyzing the representation of the object in the frames of image data to detect change over time in at least one parameter selected from the list consisting of: rotation state, scale state, differential scale state, and X-Y offset; and

controlling some aspect of the system in accordance with said detected change.

B22. The method of claim B21 in which:

the object is digitally-encoded;

the method includes decoding plural-bit digital data from the object; and

the system is responsive to at least some of said decoded digital data.

B23. The method of claim B21 in which the system responds to at least some of said decoded digital data by setting a level of sophistication of the system-user interaction accordingly.

B24. The method of claim B21 which includes analyzing the representation of the object in the frames of image data to detect change over time in rotation state, and controlling some aspect of the system in accordance with said detected change.

B25. The method of claim B21 which includes analyzing the representation of the object in the frames of image data to detect change over time in scale state, and controlling some aspect of the system in accordance with said detected change.

B26. The method of claim B21 which includes analyzing the representation of the object in the frames of image data to detect change over time in differential scale state, and controlling some aspect of the system in accordance with said detected change.

B27. The method of claim B21 which includes analyzing the representation of the object in the frames of image data to detect change over time in X-Y offset, and controlling some aspect of the system in accordance with said detected change.

B28. The method of claim B21 which includes analyzing the representation of the object in the frames of image data to detect a first change over time in one of said parameters, and thereafter to detect a second change over time in one of said parameters, and controlling some aspect of the system in accordance with said detected changes.

B29. The method of claim B21 in which the first and second changes both relate to the same parameter.

B30. In a method of operating a telephone-like appliance that establishes a voice circuit between a caller and a recipient, an improvement comprising: presenting to an optical sensor a substrate having indicia thereon, imaging a field of view that includes at least part of the substrate and producing image data corresponding thereto, steganographically decoding plural-bit digital data from the image data, and establishing the voice circuit in accordance with at least some of said plural-bit digital data.

B31. The method of claim B30 in which the indicia comprises a photographic image depicting the recipient.

B32. A method comprising:

positioning an identity object to be at least partially within the field of view of an optical scanning device, the identity object being selected from the list consisting of: a driver's license, a government identity document, a bank card, and an item of signet jewelry;

with the scanning device, grabbing a frame of image data;

processing the image data to steganographically decode plural-bit data therefrom; and

providing at least some of said steganographically decoded data to a computer system.

B33. The method of claim B32 that further includes obtaining email data from an email account corresponding to a person identified by the identity object, and providing same to a user of the computer system.

B34. An email data terminal for use in public places, the terminal including an input and an output, the output providing email data to users thereof, an improvement wherein the input includes an optical scanning device for generating scan data corresponding to an identity object, the identity object being selected from the list consisting of: a driver's license, a government identity document, and bank card.

B35. A promotional method comprising:

presenting an object within the field of view of an optical sensor device, the object being selected from the list consisting of a retail product, or packaging for a retail product;

acquiring optical data corresponding to the object;

decoding plural-bit digital data from the optical data;

submitting at least some of said decoded data to a remote computer; and

determining at the remote computer whether a prize should be awarded in response to submission of said decoded data.

B36. A method of tailoring the desktop of a first computer to conform to a user's preferences, comprising:

presenting an identity object to an optical sensor associated with the first computer;

analyzing image data produced from the optical sensor to decode plural-bit data therefrom; and

tailoring the computer desktop in accordance with said decoded data.

B37. The method of claim B36 that further includes:

using at least part of said decoded data in communicating from said first computer to a remote computer, the remote computer having profile data stored thereon relating to the user's preferred desktop arrangement;

transferring said profile data from the remote computer to the first computer; and

arranging the desktop on the first computer in accordance with said profile data.

B38. A method comprising:

purchasing a product or service from a first vendor;

receiving evidence of the purchase;

presenting the evidence to an optical scanning device to produce image data;

decoding plural-bit data from the image data;

employing said decoded plural-bit data to derive value from a second vendor unrelated to the first.

B39. The method of claim B38 in which the value is the provision of a free, or discounted, product or service from the second vendor.

B40. A method comprising:

presenting a travel document to an optical scanning device, the travel document having plural-bit data encoded thereon, the optical scanning device providing image data corresponding thereto;

discerning the plural-bit data from the image data; and

based on said plural-bit data, establishing a link to a remote computer, and employing said computer to alter travel plans.

B41. A communications method comprising:

presenting a business card to an optical sensor, the optical sensor producing image data, the business card corresponding to a person;

discerning from the image data plural-bit digital information;

using said discerned digital information to access a remote data store;

obtaining from the data store updated contact information for the person; and

establishing communication with the person using said updated contact information;

wherein the contact information at the remote data store is updated during the day as the person's activities make different contact information appropriate at different times.

B42. The method of claim B41 in which the plural-bit digital information is steganographically encoded on the business card.

B44. A method comprising:

presenting a physical object within the field of view of an optical sensor associated with a first computer, the optical sensor providing image data;

decoding plural-bit digital data from the representation of the physical object within said image data;

using a first part of said plural-bit digital data to establish a link between the first computer and a remote computer;

presenting a second part of said plural-bit digital data to the remote computer; and

providing information from the remote computer to the first computer, said information corresponding to said physical object.

B45. The method of claim B44 which includes using the first part of said plural-bit digital data to traverse a network of plural server computers to thereby identify the remote computer.

B46. The method of claim B44 in which the object is a business card relating to a person, and the information provided from the remote computer includes contact information relating to said person.

B47. A method comprising:

steganographically encoding a print advertisement with auxiliary data;

publishing said advertisement;

charging a fee for said publishing to an advertiser;

providing technology by which the steganographic encoding can be exploited to provide enhanced value to the advertiser; and

sharing a portion of said fee with a provider of said technology.

B48. A method comprising:

steganographically embedding a plural-bit digital identifier in a photograph;

storing an annotation in a database in association with said identifier;

presenting the photograph to an optical sensor, the optical sensor providing image data

decoding the identifier from the image data;

using the decoded identifier to access the annotation stored in the database; and

presenting the accessed annotation.

B49. A method of navigating the internet comprising:

linking to an internet address using a computer, and presenting information to a user of the computer from said address;

presenting an object to an optical sensor associated with the computer, the sensor producing image data corresponding to a field of view;

physically manipulating the object within said field of view;

discerning the manipulation of the object from the image data and producing output data corresponding thereto; and

changing the internet address in accordance with said output data, to thereby change the information presented to the user.

B50. A note pad comprising several sheets removably joined to each other at one edge by adhesive, characterized in that each sheet is encoded with plural-bit digital data.

B51. The note pad of claim B50 in which each sheet is steganographically encoded with plural-bit digital data.

B52. The note pad of claim B50 in which each sheet is encoded with the same plural-bit digital data as other sheets in the pad.

B53. A method comprising:

presenting an object to an optical sensor, the optical sensor producing image data, the object being steganographically encoded with plural bits of digital data;

decoding said digital data from the image data; and

in response to said digital data, storing image data from said sensor depicting the object.

B54. The method of claim B53 in which the object is a sheet of paper and the method includes

capturing several frames of image data from the optical sensor;

identifying one of said frames as having a suitable representation of the object;

masking image data corresponding to the sheet from other image data in the identified frame;

processing the image data corresponding to the sheet to change a geometrical attribute thereof;

discerning the size and color of the sheet from certain of said digital data decoded therefrom; and

storing the masked, processed image data corresponding to the sheet in association with data indicating the date and time at which the image was stored, together with the size and color of the sheet.

B55. A method comprising:

presenting an object to an optical sensor, the optical sensor producing image data, the object being steganographically encoded with plural bits of digital data;

decoding said digital data from the image data; and

in response to said digital data, displaying an image of a like object.

B56. The method of claim B55 in which the object is a sheet of paper, and the method includes displaying images of plural sheets of paper earlier stored, in a display order corresponding to the temporal order in which they were earlier stored.

B57. In a computer operating system, a software module responsive to successive frames of two-dimensional image data to extract gesture data therefrom, the gesture data including at least two data from the list consisting of: X-Y position data, rotation state, scale state, and differential scale state.

C1. A communications method comprising:

presenting a business card to an optical sensor, the optical sensor producing image data, the business card corresponding to a person;

discerning from the image data plural-bit digital information;

using said discerned digital information to access a remote data store;

obtaining from the data store updated contact information for the person; and

establishing communication with the person using said updated contact information;

wherein the contact information at the remote data store is updated during the day as the person's activities make different contact information appropriate at different times.

C2. The method of claim C1 in which the plural-bit digital information is steganographically encoded on the business card.

C3. A method comprising:

presenting a physical object within the field of view of an optical sensor associated with a first computer, the optical sensor providing image data;

decoding plural-bit digital data from the representation of the physical object within said image data;

using a first part of said plural-bit digital data to establish a link between the first computer and a remote computer;

presenting a second part of said plural-bit digital data to the remote computer; and

providing information from the remote computer to the first computer, said information corresponding to said physical object.

C4. The method of claim C3 which includes using the first part of said plural-bit digital data to traverse a network of plural server computers to thereby identify the remote computer.

C5. The method of claim C3 in which the object is a business card relating to a person, and the information provided from the remote computer includes contact information relating to said person.

D1. A method comprising:

presenting a digitally encoded object to an optical sensor, the optical sensor producing output data;

decoding plural-bit data from the sensor output data; and

using said plural-bit data to establish a link to an internet address having data relating to said object.

D2. The method of claim D1 in which the object is steganographically encoded with said plural-bit data.

E1. A promotional method comprising:

steganographically encoding a print advertisement to hide plural-bit data therein;

processing the print advertisement to extract the plural-bit data therefrom;

using at least a part of the extracted plural-bit data to direct an internet web browser to a web site that provides consumer information related to a product or service promoted by the print advertisement.

E2. A method of determining consumer response to print advertising, comprising:

steganographically encoding a first print advertisement with first data;

steganographically encoding a second print advertisement with second data;

decoding the first and second data when consumers present the first and second advertisements to an optical sensor; and

tallying the number of decoded first and second data, respectively, to determine consumer response to the advertisements.

E4. A method of travel promotion, comprising:

steganographically encoding a travel photograph to hide plural-bit data therein;

processing the travel photograph to extract the plural-bit data therefrom;

using at least part of the extracted plural-bit data to direct an internet web browser to a web site that provides travel information useful to a consumer who wishes to visit the location depicted in the photograph.

F1. In a mouse having X- and Y- position encoders, and circuitry responsive thereto for generating X- and Y- movement data and providing same to an associated computer, an improvement comprising an optical sensor disposed in said mouse, and circuitry responsive thereto for producing grey-scale optical image data and for providing same to the associated computer, wherein the mouse serves both as a positioning device and as an optical input device.

F2. A semiconductor comprising:

a substrate having formed thereon:

a two dimensional image sensor for producing sensor signals;

first circuitry responsive to said sensor signals for producing image data corresponding thereto, and for providing said image data to an image data output of said semiconductor;

second circuitry responsive to said sensor signals or to said image data for steganographically decoding plural-bit digital data therefrom, and for providing same to a steganographic decoder output of said semiconductor.

F3. In a method of operating a computer system that includes an internet web browser, an improvement comprising:

providing a peripheral device having a sensor;

positioning the peripheral device adjacent a first object to direct the web browser to a first address; and

positioning the peripheral device adjacent a second object to direct the web browser to a second address.

G1. A peripheral device for a computer system comprising:

a housing adapted to fit within a user's palm and slide over a medium;

an optical sensor having plural sensing elements and producing image signals;

a lens for imaging the medium onto the sensor;

circuitry coupled to the sensor and disposed within the housing for processing the signals from the sensor and formatting same as frames of output data; and

transfer means for relaying the output data from the peripheral device to the computer system.

G2. The device of claim G1 in which the transfer means is a cable.

G3. The device of claim G1 in which the transfer means is a wireless link.

G4. The device of claim G1 in which the circuitry decodes plural bit binary data that is steganographically encoded within image sensed by the sensor, and the transfer means relays the decoded data to the computer system.

G5. An electronic commerce method comprising:

providing a printed catalog that includes an image of an article offered for sale by a merchant, wherein the image is steganographically encoded with plural-bit binary data;

optically sensing the image to produce image data corresponding thereto;

decoding the steganographically encoded data from the image data; and

electronically ordering the article from the merchant by use of said decoded data, wherein said ordering makes use of earlier-stored customer profile information.

G6. The method of claim G5 in which the customer profile information includes clothing size data.

G7. The method of claim G6, further comprising:

processing the decoded data for transmission to a remote merchant computer, said processing including supplementing the decoded data with supplemental data corresponding to the customer;

transmitting the processed data to the remote merchant computer;

receiving first order data from the remote merchant computer, responsive to the transmitted processed data;

presenting the first order data to the customer;

receiving from the user further input selecting among options included in the first order data; and

transmitting said further input to the remote merchant computer.

G8. The method of claim G7 in which the supplemental data includes the customer profile information.

G9. The method of claim G7 in which the supplemental data includes data identifying the customer.

G10. An access control device comprising:

a badge;

a photograph on the badge;

the photograph having plural-bit auxiliary data hidden therein, said auxiliary data corresponding to an authorized bearer of the badge;

the badge further including further bearer-identification information, said further information being machine-detectable.

G11. The device of claim G10 in which the further information is encoded by an RF proximity circuit formed with the badge.

G12. The device of claim G10 in which the further information is encoded in a bar code on the badge.

G13. The device of claim G10 in which the further information is encoded in a magnetic stripe on the badge.

G14. A method of accessing a secured premise comprising:

presenting a badge to a sensor station;

generating optical data corresponding to a photograph on the badge;

processing the optical data to extract plural-bit auxiliary data steganographically encoded therein;

machine-sensing the ID of the badge from data not encoded in the photograph; and

checking correspondence between the machine-sensed ID and the auxiliary data extracted from the photograph to determine whether access to the secure premise is authorized.

I1. An image-based navigation system, comprising:
an optical input device;
a client application responsive to digital data encoded in an image; and
a router responsive to at least a part of said digital data.

I2. A computer storage medium having instructions thereon for causing a camera-equipped computer system to engage in a training exercise with a user to familiarize the user with operation of a camera-based internet navigation system.

I3. An internet navigation device comprising a substrate having an image printed thereon, the image having first and second data encoded therewith, the first data comprising an owner code, the second data comprising a router code, said device being useful in linking a camera-equipped computer system to an internet address.

I4. The device of claim I3 in which the owner and router codes are steganographically encoded in the image.

I6. An image-based network navigation method permitting a user to link to a remote computer, comprising:
detecting encoded data from a printed image;
linking to the remote computer through a network in accordance with said encoded data; and
providing an identity of the image to the remote computer.

I7. In an internet navigation system comprising a client application in communication with a remote router, a method comprising:
providing first data from the client application to the router;
providing plural data from the router to the client application;
at the client application, presenting a menu corresponding to said plural data; and
accepting user input corresponding to one of said plural data.

I8. The method of claim I7 that includes deriving the first data from image data provided from an optical input device.

I9. The method of claim I8 that includes linking the client application to a web page corresponding to the accepted user input.

I10. The method of claim I9 that includes presenting to a user a branded display locally stored at the client application while waiting for the linked web page to display.

I11. A router that accepts a routing code and responds with a URL corresponding thereto, characterized in that at least one routing code corresponds to plural URLs, each of which has temporal information detailing when the URL is to be used.

J2. A database method comprising:

generating a database record including plural data fields;

generating a file corresponding to said database record and including data from at least certain of said fields;

electronically distributing a copy of the file to each of plural recipients;

one of said recipients adding data to a copy of the file, or changing data in a copy of the file, and sending the file to the database;

updating the database record in accordance with said changed file;

generating a new file corresponding to the updated database record and including data from at least certain of said fields; and

electronically distributing a copy of the new file to each of said plural recipients.

J3. A system for linking from physical or digital objects to corresponding digital resources, comprising:

registration means for receiving data relating to an object, including its identity and owner, and associating same in a database with data relating to a corresponding response;

originating device means for sensing data from an input object, processing same, and forwarding same to a routing means;

routing means for processing the processed data from the originating device means, logging information from same, and forwarding at least certain of said processed data to a product handler means; and

product handler means for providing a response to the originating device means in accordance with the information provided thereto by the routing means.

J4. The system of claim J3 in which the routing means includes means for checking information in the database.

J5. The system of claim J3 in which the registration means includes means for generating an encapsulating file and means for distributing said file to predetermined parties.

J13. A networked computer system, responsive to watermark data sent from a remote client application to initiate delivery of audio or video data.

J14. The networked computer system of claim J13, responsive to watermark data sent from a software program on a remote computer, to initiate delivery of audio or video data to said remote computer.

J16. A networked computer system, responsive to watermark data sent from a software program on a remote computer, to initiate delivery of updated software to said remote computer.

K1. In a scanner including a CPU, a memory, a linear sensor array, and first and second spaced-apart 2D sensor arrays serving as motion encoders, the CPU serving to process raw scan data collected by the linear sensor array from an imaged object into final scan data in accordance with scanner motion data provided from said 2D sensors, an improvement comprising software instructions in the memory causing the scanner to discern a machine-readable identifier from scan data acquired from the object.

K2. The scanner of claim K1, further comprising a wireless interface, and a display, said software instructions causing the scanner to relay the identifier by the wireless interface to a remote server, and for processing information returned through the wireless interface for presentation on said display.

K3. The scanner of claim K1 in which the software instructions cause the CPU to process data from the 2D sensor arrays for a purpose in addition to sensing scanner motion.

K4. The scanner of claim K3 in which said purpose includes beginning a watermark detection process before data from the linear sensor array is finally processed.

K5. The scanner of claim K4 in which said purpose includes beginning to sense a watermark calibration signal.

K6. The scanner of claim K3 in which said purpose includes identifying portions of the data collected by the linear sensor array that are relatively more likely to include detectable identifier data.

K7. The scanner of claim K3 in which said purpose is to quantify an object surface characteristic, wherein a filter can be applied to the scan data in accordance therewith.

K8. The scanner of claim K3 in which said purpose is to assess relative distance to the object from different portions of the scanner.

K9. The scanner of claim K3 in which said purpose is to quantify an affine distortion in the scan data, so that compensation may be applied therefore.

K10. The scanner of claim K1 in which the identifier is steganographically encoded as a digital watermark.

K11. The scanner of claim K1 in which the identifier is encoded as a barcode.

K12. A scanner comprising a linear sensor array, a CPU, an interface for coupling to the internet, and a user interface including a display screen and user controls, the scanner being adapted to transmit machine-readable data sensed from scanned objects to the internet, and for presenting to a user HTML information received back from the internet.

L1. In an image watermarking technique that includes changing an optically-sensible pattern on an object, an improvement wherein the object has a non-planar surface, and the watermark applied to said surface is pre-warped to precompensate for said non-planarity.

L2. The method of claim L1 in which the watermark is geometrically pre-warped.

L3. The method of claim L1 in which the object has a cylindrical surface.

L4. The method of claim L3 in which the object is a container for consumer goods.

L5. The method of claim L4 in which the object is a soft drink can.

L6. The method of claim L1 in which the watermark is pre-warped in a transform domain.

L7. The method of claim L6 in which the watermark is pre-warped in the wavelet domain.

L8. The method of claim L6 in which the watermark is pre-warped in the DCT domain.

L9. The method of claim L1 in which the watermark is pre-warped to account for a curvature-induced apparent geometrical distortion.

L10. The method of claim L1 in which said surface is pre-warped to account for a distance-induced apparent geometrical distortion.

L11. The method of claim L10 in which the watermark is pre-warped to account for a curvature-induced apparent geometrical distortion.

L12. In watermark decoding technique that includes analyzing a frame of image data to decode a watermark encoded therein, an improvement comprising applying an unwarping function to the image data prior to decoding the watermark.

L13. The method of claim L12 which includes successively applying different unwarping functions to the image data to attempt to decode valid watermark data therefrom.

L14. The method of claim L12 which includes selecting one or more unwarping functions to apply based on information provided by a user.

L15. The method of claim L14 in which said information identifies a general class of objects to which the object belongs, without specifically identifying a specific object shape.

Conclusion

Having described and illustrated the principles of our technology with reference to illustrative embodiments, it should be recognized that the invention is not so limited.

For example, while certain of the embodiments were illustrated with reference to internet-based systems, the same techniques are similarly applicable to any other computer-based system. These include non-internet based services such as America Online and Compuserve, dial-up bulletin board systems, etc. Likewise, for internet-based embodiments, the use of web browsers and web pages is not essential; other digital navigation devices and other on-line data repositories can be similarly accessed.

Similarly, while the details of illustrative systems were particularly given, the underlying principles can be employed in numerous other forms.

For example, one other form is to steganographically encode physical objects with Digital Object Identifiers (DOIs). The Center for National Research Initiatives and the Digital Object Identifier Foundation (www.doi.org) have performed extensive work in establishing an infrastructure by which digital objects can be distributed, tracked, and managed. Some of this same infrastructure and technology can be adapted, in accordance with the teachings provided above, to associate new functionality with physical objects.

Another form is not to reference a remote data repository by data embedded on an object, but instead to encode the ultimate data directly on the object. A photograph, for example, can be literally encoded with a telephone number. On presenting the photograph to an optical sensor on the telephone, the telephone can analyze the optical information to extract the telephone number, and dial the number, without the need for any external data. Similarly, a printed office document (e.g., spreadsheet) can be encoded with the path and file name of the corresponding electronic file, obviating the need for indirect

linking (e.g., to a remote database to correlate a UID to a computer address). A person's business card can directly encode that person's email or web address. Most of the above-described embodiments are suitable for such direct encoding of the related data.

In the business card example given above, the detailed techniques can be supplementary to existing optical character recognition techniques. That is, the image data from an optical sensor can be applied both to a Bedoop decoder and to an OCR system. Text characters discerned by the OCR system can be entered directly into a contacts manager personal database. The techniques employed in the Bedoop system to locate the encoded object and handle visual distortion (e.g., the visual artifacts due to scale, rotation, etc.) can advantageously be used in OCR detection as well, permitting extraction of the OCR information without careful placement of the card.

While certain of the foregoing embodiments made reference to ink-jet printing, similar advantages can often be obtained with other printing technologies, e.g., laser/xerographic printing, offset printing, etc.

In the foregoing embodiments, Bedoop decoding generally proceeded from image data obtained from a physical object. However, in some contexts, it is advantageous to Bedoop-decode image data provided electronically, e.g., over the internet.

Likewise, while the foregoing embodiments generally relied on Bedoop image sensors that stared out for an object at an expected point, in alternative embodiments, sensors that seek rather than stare can be employed (as was illustrated above in connection with the elevator example).

Similarly, while the illustrated embodiments generally employed sensors that repeatedly grabbed frames of image data, this need not be the case. Single frame systems, such as flatbed scanners, and video systems arranged to grab single frames -- with or without TWAIN interfaces -- can alternatively be used.

As indicated above, while steganographic encoding of the digital data is used in the preferred embodiments, visible forms of digital encoding – such as bar codes – can naturally be employed where aesthetic considerations permit.

In certain of the embodiments, digital data conveyed by means other than optical can be used. Electromagnetic detection (e.g., of the sort used in proximity-based card-access systems) can be arranged to decode digital data, permitting "at-a-distance" reading of data from physical objects, just as in the foregoing embodiments.

Since the Bedoop image sensors typically acquire plural frames of data, the extraction of the digital data can be based on more than a single image frame. More confidence in the results may be accumulating decoded data over several frames. Moreover, movement of the object within the sensor's field of view may permit the system to acquire information from other perspectives, etc., enhancing system operation.

While the preferred embodiments employ 2-D image sensors (e.g., CCDs), other optical sensing technology can alternatively be employed. Supermarket laser scanners, for example, can read bar-code

data. Raster-scanning of such systems can permit acquisition of 2-D data (either in bit-mapped form, or grey-scale).

Some embodiments can advantageously employ texture-based Bedoop encoding of objects. Bedoop texturing can be effected by various means, including pressure rollers, chemical or laser etching, etc.

It should be noted that the responses triggered by a watermark can be changed over time. This extends the useful life of an encoded object. For example, an encoded link in a magazine ad for a 1999 Ford Explorer that points to a Ford URL related to the 1999 Ford Explorer can be updated to point to the URL for the 2000 model year version when marketing of the new model year vehicles commences.

In other embodiments, of course, a literal URL can be encoded, and can be used to direct a browser or other information appliance to that address. Still further, a literal URL can be encoded, but not necessarily used. Instead, the encoded URL can be mapped to an actual URL (i.e., a URL to which the browser is directed or to which a response to the MediaBridge object is otherwise keyed) through a database. In one such embodiment, a URL is encoded in an object together with a future date. On "reading" the object, the local (client) computer checks the associated date. If the date has not passed, the literal URL is used as the actual URL. If the date has passed, the client computer refers to the code to a remote database (e.g., at the router) to obtain an actual URL (which may be the same – if no update has been required, or may be new). The actual URL is then used in providing a response to the MediaBridge object.

The implementation of the watermark encoding and decoding systems is straightforward to artisans in the field, and thus not unduly belabored here. Conventionally, such technology is implemented by suitable software, stored in long term memory (e.g., disk, ROM, etc.), and transferred to temporary memory (e.g., RAM) for execution on an associated CPU. In other implementations, the functionality can be achieved by dedicated hardware, or by a combination of hardware and software. Reprogrammable logic, including FPGAs, can advantageously be employed in certain implementations.

While the foregoing embodiments have generally employed planar objects to convey the digital encoding, this need not be the case. Objects of other shapes can likewise be employed. Some shapes present relatively straightforward image processing tasks. As noted above, data imaged from a soft drink can or other cylindrical surface can be remapped using known geometrical transforms so as to essentially "unwrap" the printing from the can. Other geometries can present more complex re-mappings, but are likewise generally within the capabilities of the artisan. (Such remapping is facilitated by encoding in the data certain reference markings, such as subliminal graticules, etc. The unknown 3D shape of the object being imaged can usually be inferred from the apparent warping of the reference markings in the 2D image data generated by the scanner. Once the warping is characterized, it is generally straightforward to un-warp so as to prepare the image data for decoding.)

It was once popular to predict that paper documents would be replaced with electronic media. In hindsight, electronic media may be recognized as a poor surrogate for paper. Electronic media conveys information flawlessly, but is lacking in experiential attributes. We can hold paper, stack it,

own it, deface it, give it, guard it, etc. It provides an opportunity for physical dominion entirely lacking with electronic media.

From the foregoing discussion it can be seen that, rather than replacing paper with electronic media, perhaps the future lies in giving paper digital attributes – hybridizing the physical experience of paper with the technical advantages of digital media. Such an arrangement makes available a great wealth of new functionality, now accessible through familiar paper items, rather than through a "computer input peripheral."

To provide a comprehensive disclosure without unduly lengthening this specification, applicants incorporate by reference the patents, applications, and publications identified above.

In view of the many embodiments to which the above-detailed principles may be applied, it should be recognized that the detailed embodiments are illustrative only and should not be taken as limiting the scope of our invention. Rather, we claim as our invention all such embodiments as fall within the scope and spirit of the following claims, and equivalents thereto.

WE CLAIM

1. A method of data processing on a computer system, comprising:

using an application program to compose an electronic version of a text-based document;

printing the document onto paper, the printing including steganographically encoding the printed paper with plural-bit auxiliary data;

storing the plural-bit auxiliary data in association with data identifying a location at which the electronic version of the document is stored.

2. A method of data processing on a computer system, comprising:

presenting a text-based printed document to an optical capture device;

processing image data produced by said device to decode plural-bit data steganographically encoded therein;

based on said decoded plural-bit data, launching a software application corresponding to said printed document; and

using said software application to open an electronic version of said document.

3. In a method of operating a computer, the computer including an operating system with a registry database, the registry database associating specific data types with specific software programs particularly corresponding thereto, an improvement comprising:

providing a frame of image data;

decoding plural-bit identifier data from the image data;

consulting the registry database to identify a software program corresponding to said identifier data; and

invoking the identified software program.

4. The method of claim 3 which includes:

steganographically decoding plural fields of data from the image data, one of said fields comprising the identifier data; and

providing another of said fields of data to the identified software program for use thereby.

5. A greeting card comprising a substrate with visually-perceptible indicia printed thereon, wherein the card is steganographically encoded with plural-bit binary data that can be decoded by an image processing device and used to direct a computer to a web site where an image, video, and/or audio presentation corresponding to said card is provided.

6. A method of providing a customized greeting, comprising:

acquiring a greeting card having plural-bit data steganographically encoded therein;

customizing a web site presentation corresponding to said card;

providing the card to a recipient;

decoding the steganographically encoded plural-bit data from the card; and

in response to said steganographically decoded plural-bit data, presenting to the recipient the web site presentation.

7. A method of printing a magazine, comprising:

processing an electronic representation of an advertisement with a digital watermark to steganographically encode plural bit data therein;

printing a page of advertising in accordance with said electronic representation to yield a steganographically encoded advertisement page; and

binding said page into a magazine;

wherein said plural bit data serves to identify an entry in a database, said database entry having stored therein an internet address of a web page that is associated with said advertisement.

8. A promotional method comprising:

steganographically encoding a print advertisement to hide plural-bit data therein;

processing the print advertisement to extract the plural-bit data therefrom; and

using at least a part of the extracted plural-bit data to direct an internet web browser to a web site that provides consumer information related to a product or service promoted by the print advertisement.

9. A method of determining consumer response to print advertising, comprising:

encoding a first print advertisement with first data;

encoding a second print advertisement with second data different than the first;

the first and second data providing identifiers by which consumer devices can link to web pages associated with said advertisements;

monitoring linking traffic due to each of said identifiers to thereby determine consumer response to the advertisements.

10. The method of claim 9 in which the encoding takes the form of visible bar codes.

11. The method of claim 9 in which the encoding takes the form of steganographic digital watermarks.

12. The method of claim 9 in which the first and second advertisements are identical, except for the data encoded therein.

13. The method of claim 12 in which the first and second advertisements are published in different magazines.

14. The method of claim 9 in which the first and second advertisements are different, apart from the data encoded therein.

15. The method of claim 14 in which the first and second advertisements are published in the same magazine.

16. A promotional method comprising:

presenting an object within the field of view of an optical sensor device, the object being selected from the list consisting of a retail product, packaging for a retail product, or printed advertising;

acquiring optical data corresponding to the object;

decoding plural-bit digital data from the optical data;

submitting at least some of said decoded data to a remote computer; and

determining at the remote computer whether a prize should be awarded in response to submission of said decoded data.

17. A method of interacting with a magazine using a computer, the computer including an internet web browser, the method including:

providing a peripheral device having a sensor;

positioning the peripheral device adjacent a first advertisement in the magazine to direct the web browser to a first internet address; and

positioning the peripheral device adjacent a second advertisement in the magazine to direct the web browser to a second internet address.

18. The method of claim 17 in which the positioning comprises scanning barcode indicia published in said advertisements.

19. The method of claim 17 in which the peripheral device is a camera device having a sensor with rows and columns of sensor elements.

20. A computer peripheral and method of its use, the peripheral being used in conjunction with a computer system having an internet browser associated therewith, the peripheral comprising:

a housing adapted to fit within a user's palm and slide over a medium;

an optical sensor having at least one sensing element and producing optical data;

a lens for imaging the medium onto the sensor;

the method comprising:

sliding the peripheral over a portion of a printed advertisement;

processing the optical data to decode plural bit information encoded on the advertisement; and

using said plural bit information to direct the internet browser to an internet web page associated with said advertisement.

21. The arrangement of claim 20 in which the plural bit information does not comprise a web address, but rather comprises an advertisement identifier, and in which the method further includes:

transmitting said advertisement identifier to a remote computer;

receiving from said remote computer a web address associated with said identifier; and

directing the internet browser on said computer system in accordance with said web address.

22. The arrangement of claim 20 which includes sliding the peripheral over a portion of the printed advertisement having a barcode printed thereon.

23. An electronic commerce method comprising:

providing a printed catalog that includes an image of an article offered for sale by a merchant, wherein the image is encoded with plural-bit binary data;

optically sensing the image to produce optical data corresponding thereto;

decoding the encoded data from the optical data; and

electronically ordering the article from the merchant by use of said decoded data, wherein said ordering makes use of earlier-stored customer profile information.

24. The method of claim 23 in which the customer profile information includes clothing size data.

25. In a wireless telephony handset including a microphone, a modulator, and an RF amplifier, the device serving to receive audio and transmit an RF signal conveying audio modulation, an improvement comprising an optical sensor producing optical data, a lens for imaging an object onto the sensor, and a decoder for decoding plural bit identifier data conveyed by a barcode or a digital watermark on the object.

26. An image-based network navigation method permitting a user to link to a remote computer, comprising:

detecting encoded data from a printed object;

linking to the remote computer through a network in accordance with said encoded data; and

providing the user's zip code to the remote computer.

27. A method comprising:

sensing an object identifier from a first object;

sending said first object identifier from a first device to a second device;

in response, at said second device, identifying address information corresponding to said first object identifier and sending same to the first device;

initiating a link from the first device in accordance with said address information;

at said second device, identifying additional objects related to said first object; identifying additional address information corresponding to said additional objects; and sending said additional address information to the first device;

storing said additional address information in a memory at the first device;

wherein, if an object included among said identified additional objects is sensed by the first device, the corresponding address information can be retrieved from said memory in the first device without the intervening delays of communicating with the second device.

28. An apparatus including a detector of machine readable data and a software program used in conjunction with said machine readable data, operable to transmit a packet of data to a remote system, said packet of data comprising (a) an identifier of said software program, and (b) at least a portion of detected machine readable data.

29. The apparatus of claim 28, wherein said packet of data also includes address information identifying the apparatus.

30. The apparatus of claim 28 comprising a detector of digital watermark data.

31. The apparatus of claim 28 comprising a detector of barcode data.

32. An apparatus including a detector of machine readable data and a software program used in conjunction with said machine readable data, operable to transmit a packet of data to a remote system, said packet of data comprising (a) a context or environment identifier, and (b) at least a portion of detected machine readable data.

33. The apparatus of claim 32, wherein said packet of data also includes address information identifying the apparatus.

34. The apparatus of claim 32 comprising a detector of digital watermark data.

35. The apparatus of claim 32 comprising a detector of barcode data.

36. A networked computer system, responsive to watermark data sent from a software program on a remote computer, to initiate delivery of advertisement data to said remote computer.

37. A system as described and detailed above.

5

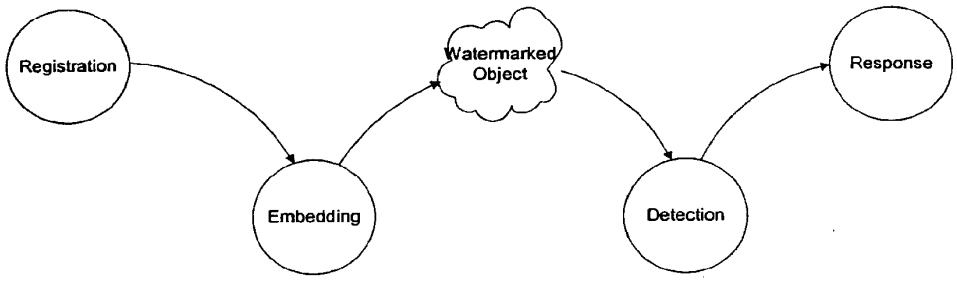

Registration
Embedding
Watermarked Object
Detection
Response

FIG. 1

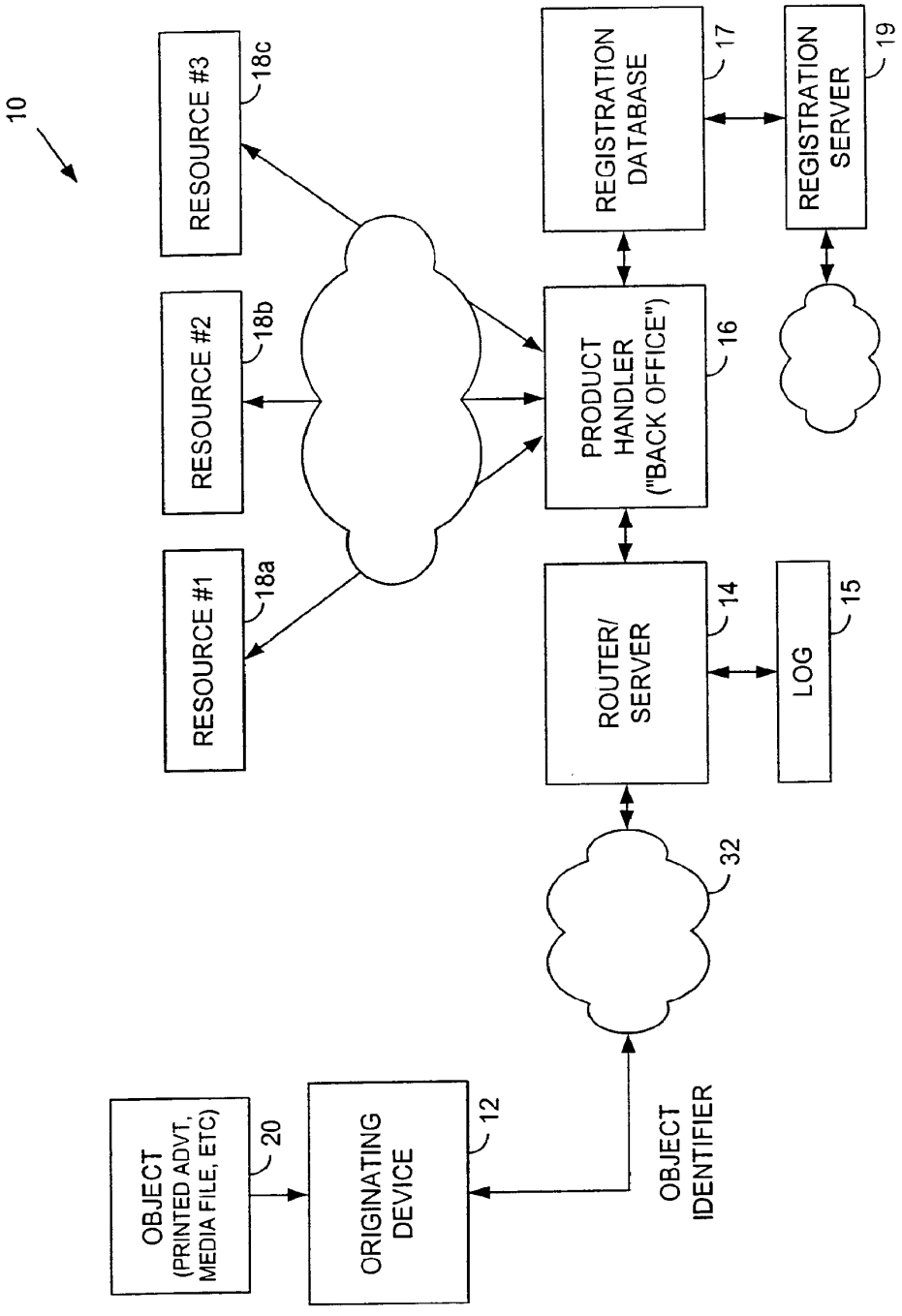

10
RESOURCE #3
18c
RESOURCE #2
18b
RESOURCE #1
18a
REGISTRATION DATABASE
17
REGISTRATION SERVER
19
PRODUCT HANDLER ("BACK OFFICE")
16
ROUTER/ SERVER
14
LOG
15
32
OBJECT (PRINTED ADVT, MEDIA FILE, ETC)
20
ORIGINATING DEVICE
12
OBJECT IDENTIFIER

FIG. 2

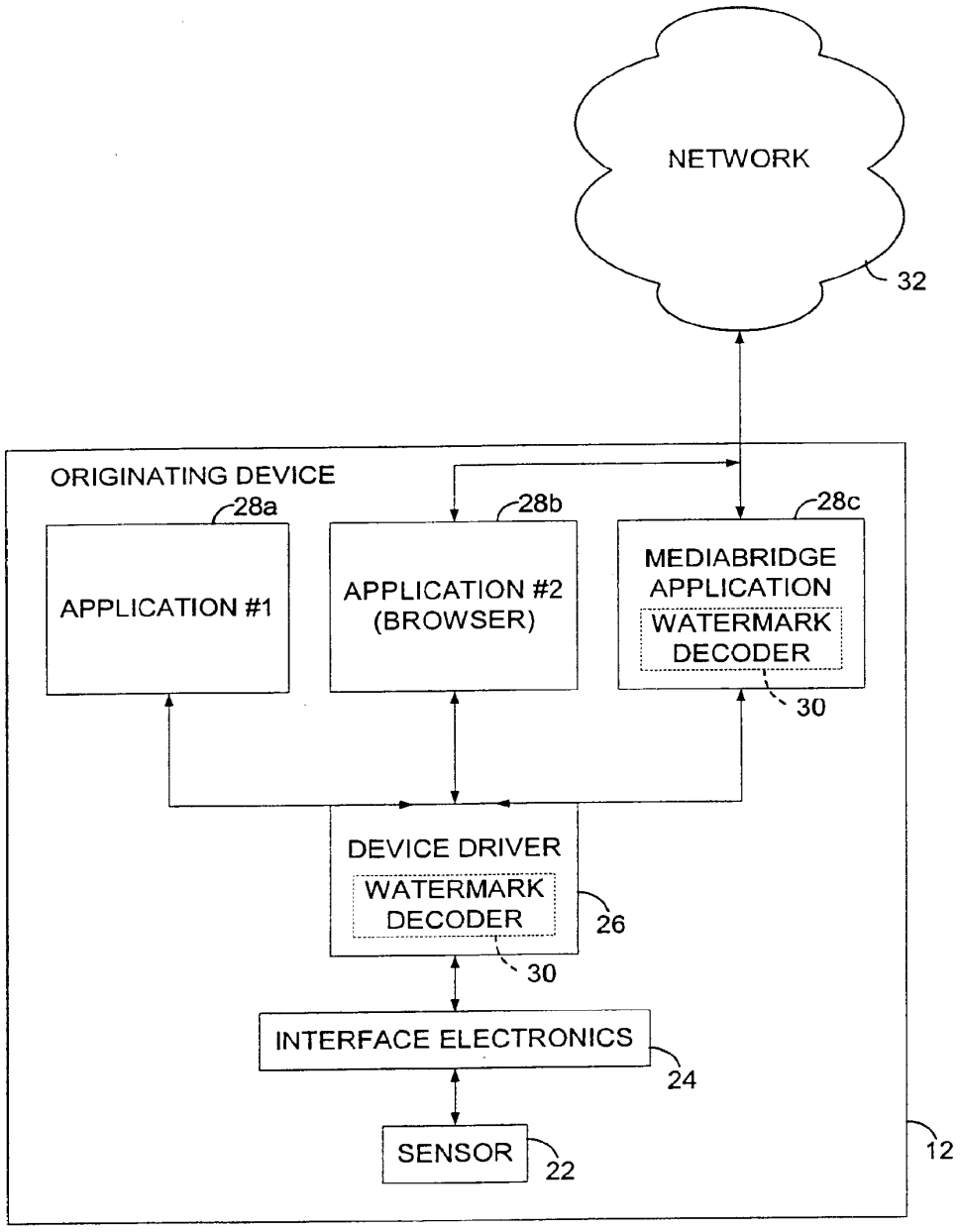

NETWORK
32
ORIGINATING DEVICE
28a
28b
28c
APPLICATION #1
APPLICATION #2
(BROWSER)
MEDIABRIDGE
APPLICATION
WATERMARK
DECODER
30
DEVICE DRIVER
WATERMARK
DECODER
26
30
INTERFACE ELECTRONICS
24
SENSOR
22
12

FIG. 3

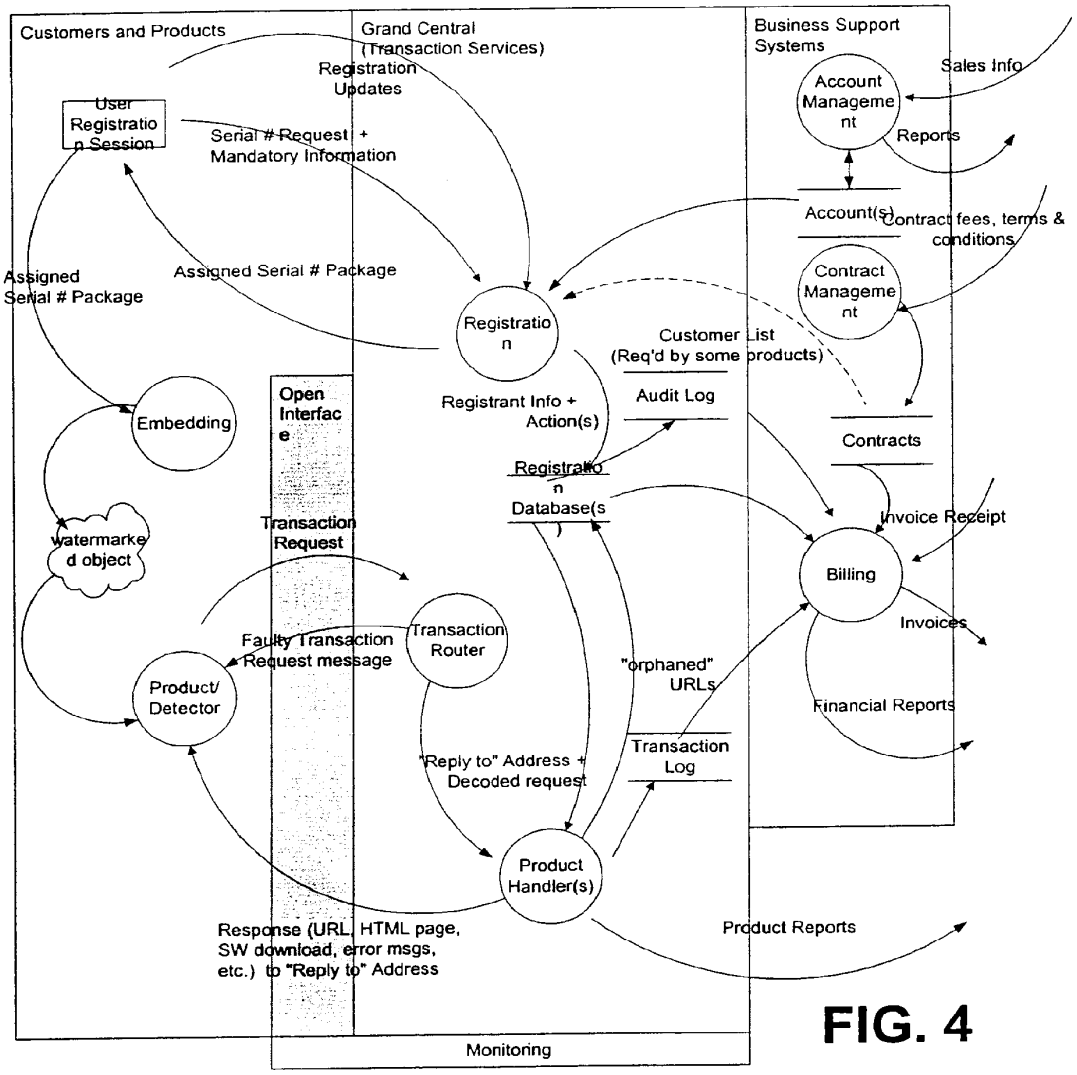

Customers and Products
Grand Central (Transaction Services)
Business Support Systems
Registration Updates
User Registration Session
Serial # Request + Mandatory Information
Assigned Serial # Package
Assigned Serial # Package
Registration
Embedding
Open Interface
watermarked object
Transaction Request
Faulty Transaction Request message
Transaction Router
Product/ Detector
"Reply to" Address + Decoded request
Product Handler(s)
Response (URL, HTML page, SW download, error msgs, etc.) to "Reply to" Address
Registrant Info + Action(s)
Registration Database(s)
Audit Log
Customer List (Req'd by some products)
"orphaned" URLs
Transaction Log
Account Management
Sales Info
Reports
Account(s)
Contract fees, terms & conditions
Contract Management
Contracts
Invoice Receipt
Billing
Invoices
Financial Reports
Product Reports
Monitoring

FIG. 4

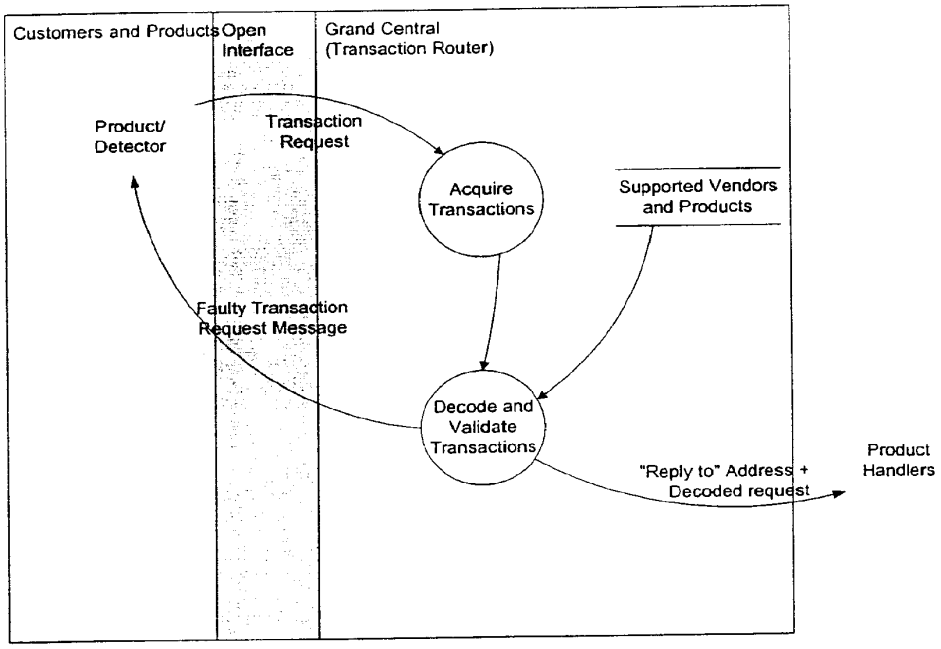

Customers and Products
Open Interface
Grand Central (Transaction Router)
Product/ Detector
Transaction Request
Acquire Transactions
Supported Vendors and Products
Faulty Transaction Request Message
Decode and Validate Transactions
"Reply to" Address + Decoded request
Product Handlers

FIG. 5

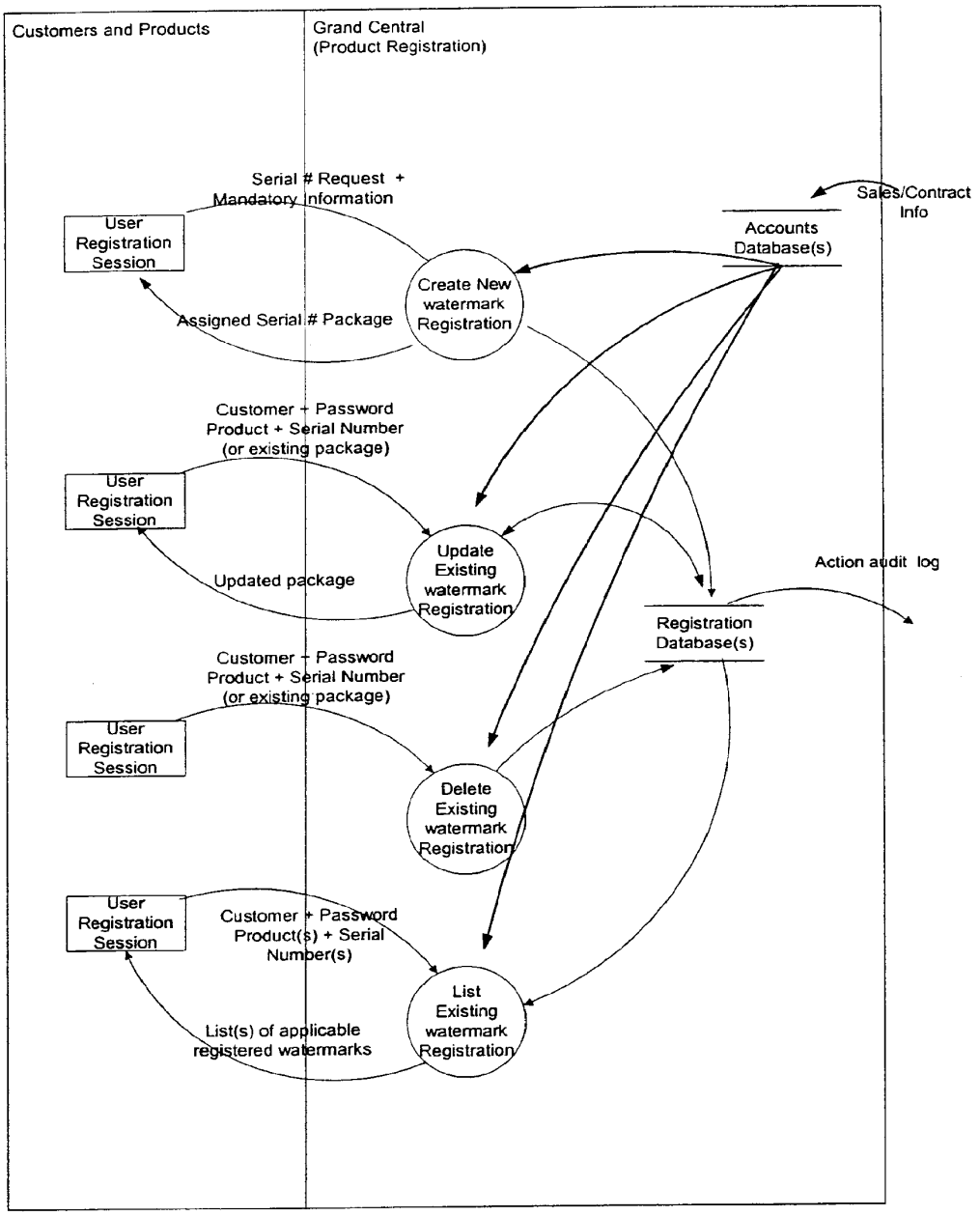

Customers and Products
Grand Central
(Product Registration)
Serial # Request +
Mandatory Information
User
Registration
Session
Assigned Serial # Package
Create New
watermark
Registration
Sales/Contract
Info
Accounts
Database(s)
Customer + Password
Product + Serial Number
(or existing package)
User
Registration
Session
Updated package
Update
Existing
watermark
Registration
Action audit log
Registration
Database(s)
Customer + Password
Product + Serial Number
(or existing package)
User
Registration
Session
Delete
Existing
watermark
Registration
User
Registration
Session
Customer + Password
Product(s) + Serial
Number(s)
List(s) of applicable
registered watermarks
List
Existing
watermark
Registration

FIG. 6

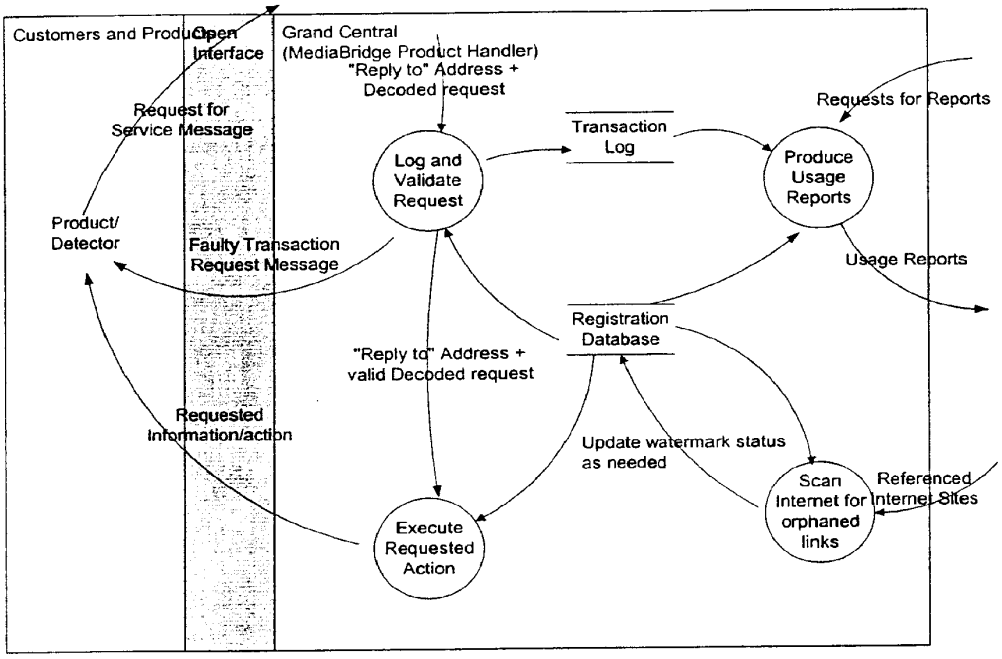

Customers and Produc
Open Interface
Grand Central
(MediaBridge Product Handler)
"Reply to" Address +
Decoded request
Request for Service Message
Transaction Log
Requests for Reports
Log and Validate Request
Produce Usage Reports
Product/ Detector
Faulty Transaction Request Message
Usage Reports
Registration Database
"Reply to" Address + valid Decoded request
Requested Information/action
Update watermark status as needed
Scan Internet for orphaned links
Referenced Internet Sites
Execute Requested Action

FIG. 7

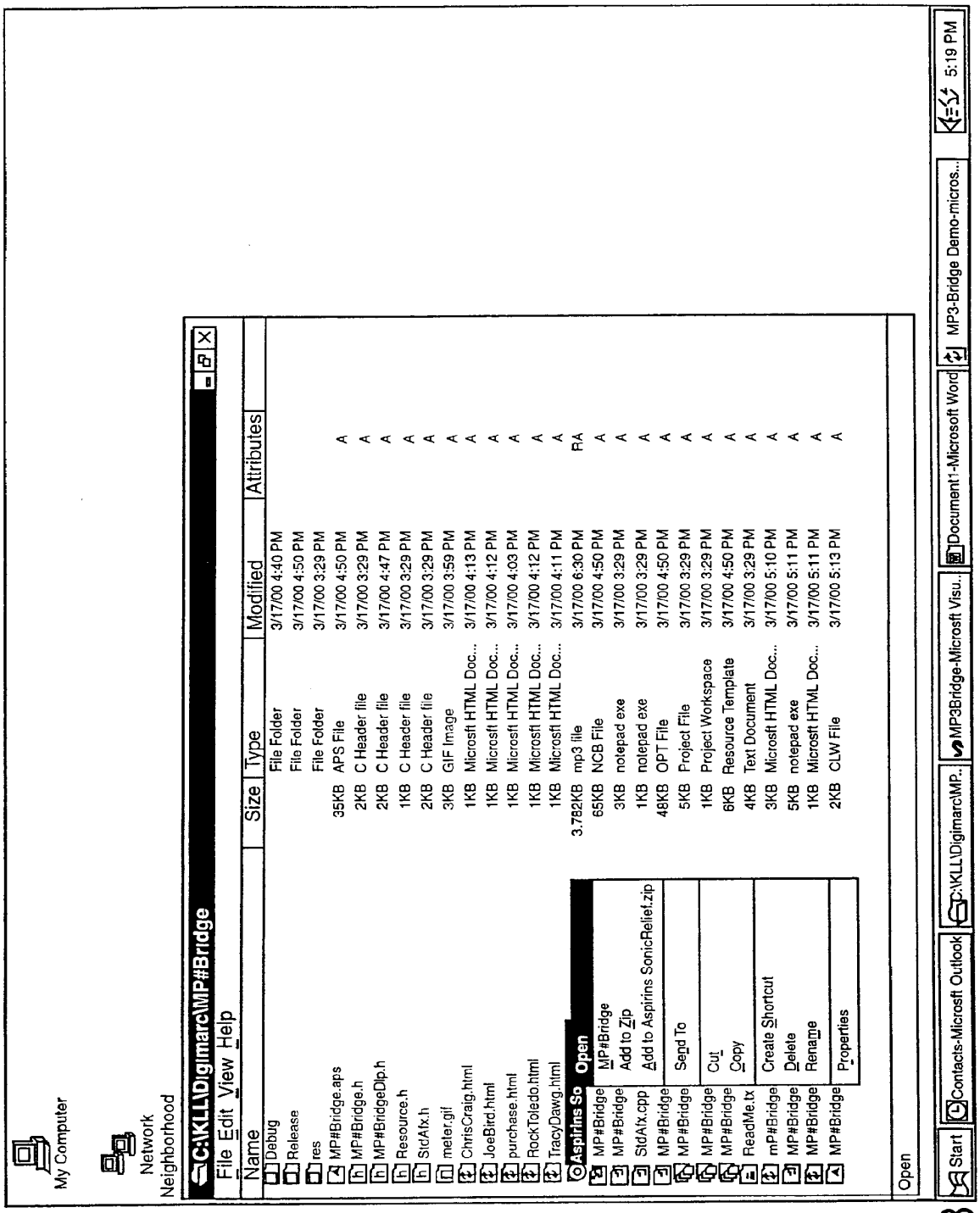

My Computer
Network Neighborhood
C:\KLL\DigimarcMP#Bridge
File Edit View Help
Name
Size
Type
Modified
Attributes
Debug
Release
res
MP#Bridge.aps
MP#Bridge.h
MP#BridgeDlg.h
Resource.h
StdAfx.h
meter.gif
ChrisCraig.html
JoeBird.html
purchase.html
RockToledo.html
TracyDawg.html
Aspirins So
Open
MP#Bridge
Add to Zip
Add to Aspirins SonicRelief.zip
Send To
Cut
Copy
Create Shortcut
Delete
Rename
Properties
File Folder 3/17/00 4:40 PM
File Folder 3/17/00 4:50 PM
File Folder 3/17/00 3:29 PM
35KB APS File 3/17/00 4:50 PM A
2KB C Header file 3/17/00 3:29 PM A
2KB C Header file 3/17/00 4:47 PM A
1KB C Header file 3/17/00 3:29 PM A
2KB C Header file 3/17/00 3:29 PM A
3KB GIF Image 3/17/00 3:59 PM A
1KB Microsft HTML Doc... 3/17/00 4:13 PM A
1KB Microsft HTML Doc... 3/17/00 4:12 PM A
1KB Microsft HTML Doc... 3/17/00 4:03 PM A
1KB Microsft HTML Doc... 3/17/00 4:12 PM A
1KB Microsft HTML Doc... 3/17/00 4:11 PM A
3,782KB mp3 file 3/17/00 6:30 PM RA
65KB NCB File 3/17/00 4:50 PM A
3KB notepad exe 3/17/00 3:29 PM A
1KB notepad exe 3/17/00 3:29 PM A
48KB OPT File 3/17/00 4:50 PM A
5KB Project File 3/17/00 3:29 PM A
1KB Project Workspace 3/17/00 3:29 PM A
6KB Resource Template 3/17/00 4:50 PM A
4KB Text Document 3/17/00 3:29 PM A
3KB Microsft HTML Doc... 3/17/00 5:10 PM A
5KB notepad exe 3/17/00 5:11 PM A
1KB Microsft HTML Doc... 3/17/00 5:11 PM A
2KB CLW File 3/17/00 5:13 PM A
Open
Start
Contacts-Microsft Outlook
C:\KLL\DigimarcMP..
MP3Bridge-Microsft Visu..
Document1-Microsoft Word
MP3-Bridge Demo-micros...
5:19 PM

FIG. 8

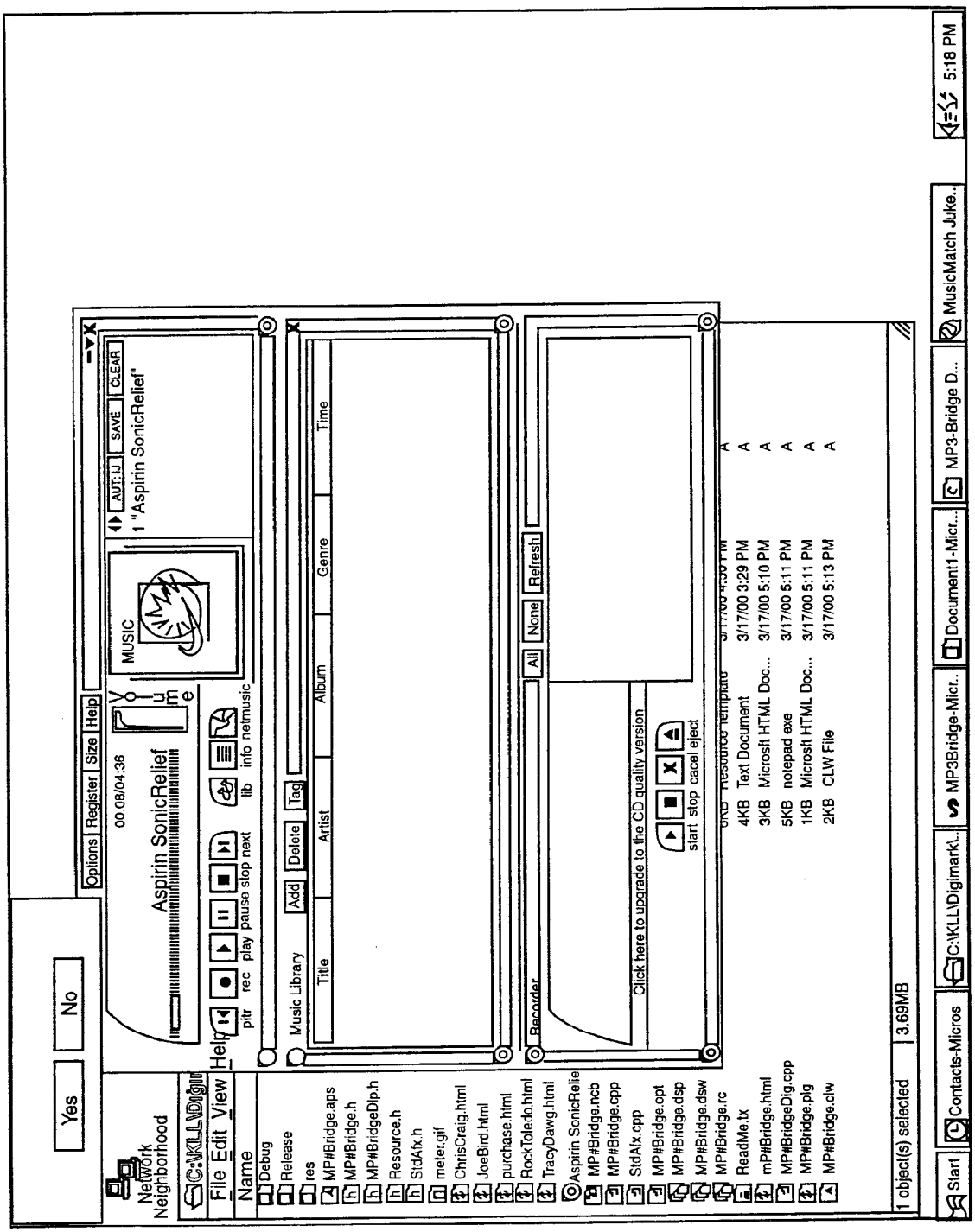

Yes
No
Network Neighborhood
C:\KLLIDigi
File Edit View
Name
Debug
Release
res
MP#Bridge.aps
MP#Bridge.h
MP#BridgeDlg.h
Resource.h
StdAfx.h
meter.gif
ChrisCraig.html
JoeBird.html
purchase.html
RockToledo.html
TracyDawg.html
Aspirin SonicRelie
MP#Bridge.ncb
MP#Bridge.opt
MP#Bridge.dsp
MP#Bridge.dsw
MP#Bridge.rc
ReadMe.tx
mP#Bridge.html
MP#BridgeDlg.cpp
MP#Bridge.plg
MP#Bridge.clw
Help
pltr rec play pause stop next lib info netmusic
Options Register Size Help
00:09/04:36
Aspirin SonicRelief
Volume
MUSIC
AUT:U SAVE CLEAR
1 "Aspirin SonicRelief"
Music Library
Add Delete Tag
Title Artist Album Genre Time
Recorder
All None Refresh
Click here to upgrade to the CD quality version
start stop cacel eject
4KB Text Document 3/17/00 3:29 PM A
3KB Microsft HTML Doc... 3/17/00 5:10 PM A
5KB notepad exe 3/17/00 5:11 PM A
1KB Microsft HTML Doc... 3/17/00 5:11 PM A
2KB CLW File 3/17/00 5:13 PM A
1 object(s) selected
3.69MB
Start
Contacts-Micros
C:\KLLIDigimarkL..
MP3Bridge-Micr..
Document1-Micr...
MP3-Bridge D...
MusicMatch Juke..
5:18 PM

FIG. 9

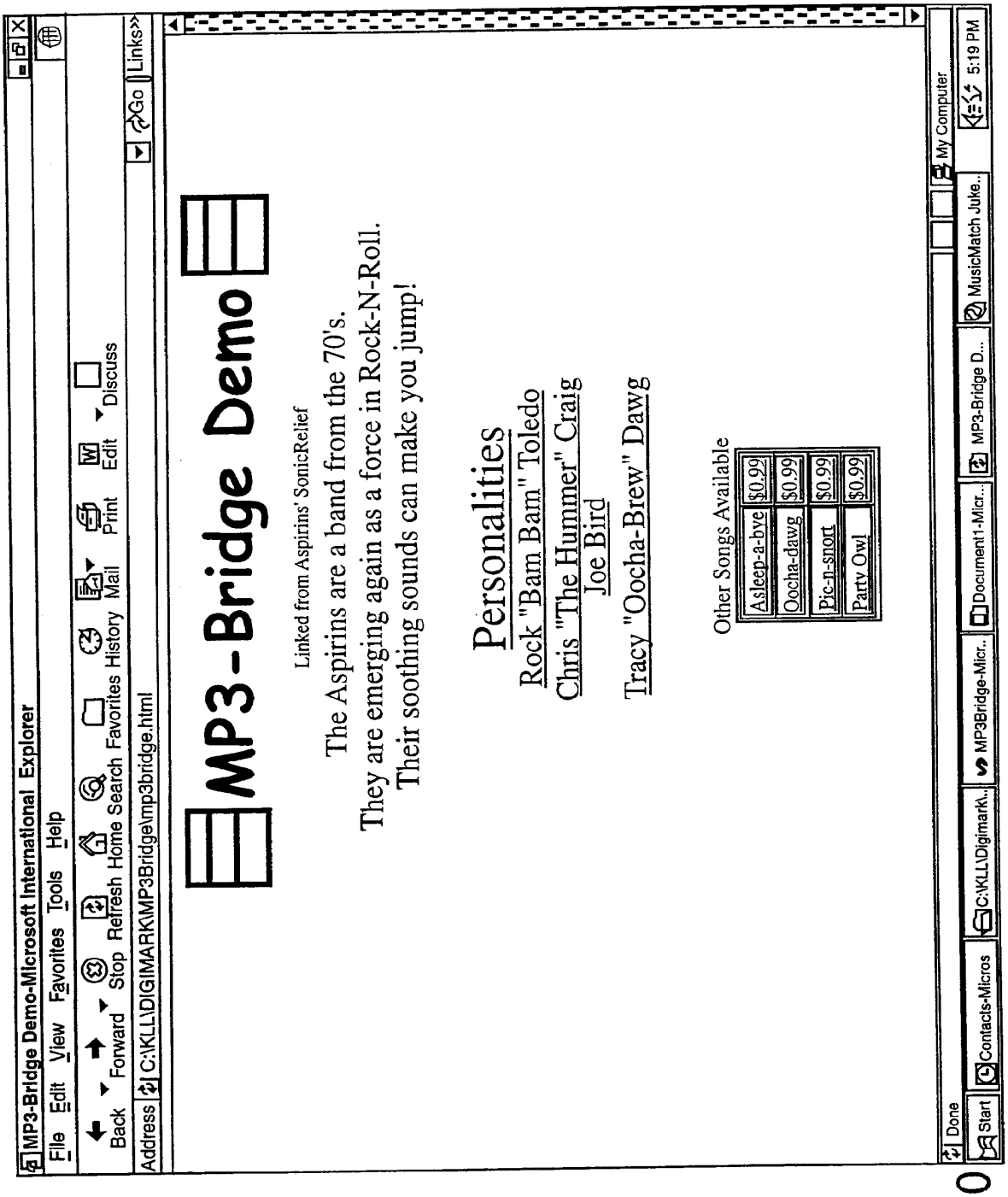

MP3-Bridge Demo-Microsoft International Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address C:\KLL\DIGIMARK\MP3Bridge\mp3bridge.html
Links
MP3-Bridge Demo
Linked from Aspirins' SonicRelief
The Aspirins are a band from the 70's.
They are emerging again as a force in Rock-N-Roll.
Their soothing sounds can make you jump!
Personalities
Rock "Bam Bam" Toledo
Chris "The Hummer" Craig
Joe Bird
Tracy "Oocha-Brew" Dawg
Other Songs Available
Asleep-a-bye $0.99
Oocha-dawg $0.99
Pic-n-snort $0.99
Party Owl $0.99
Done
My Computer
Start
Contacts-Micros
C:\KLL\Digimark\..
MP3Bridge-Micr..
Document1-Micr..
MP3-Bridge D...
MusicMatch Juke..
5:19 PM

FIG. 10

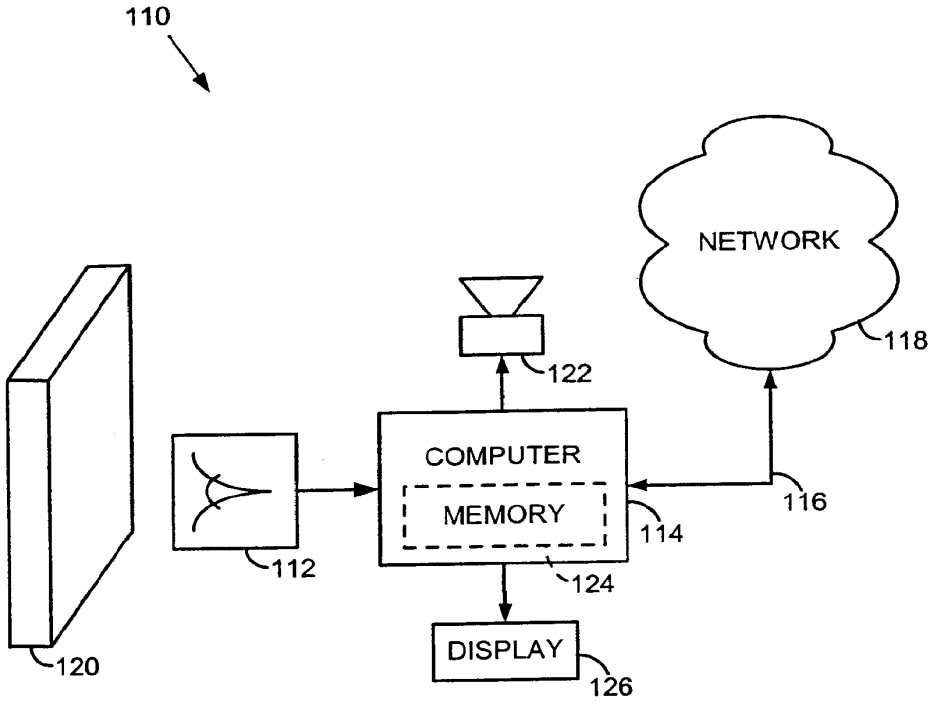

110
NETWORK
118
122
COMPUTER
MEMORY
114
116
112
124
DISPLAY
126
120

FIG. 11

FIG. 12

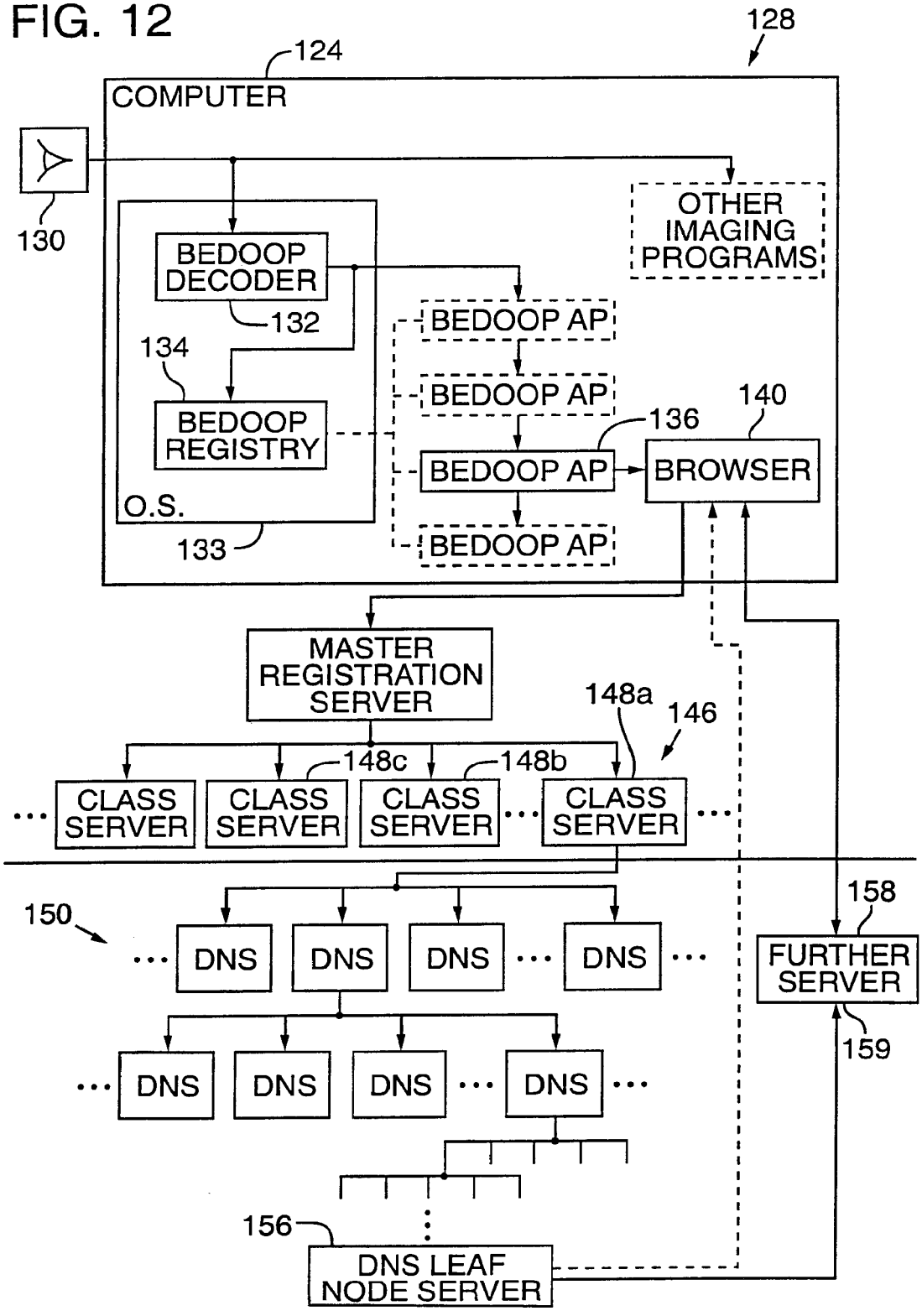

128
124
COMPUTER
130
OTHER IMAGING PROGRAMS
BEDOOP DECODER
132
134
BEDOOP REGISTRY
O.S.
133
BEDOOP AP
BEDOOP AP
BEDOOP AP
BEDOOP AP
136
140
BROWSER
MASTER REGISTRATION SERVER
148a
146
148c
148b
CLASS SERVER
CLASS SERVER
CLASS SERVER
CLASS SERVER
150
DNS
DNS
DNS
DNS
DNS
DNS
DNS
DNS
DNS
156
DNS LEAF NODE SERVER
158
FURTHER SERVER
159

FIG. 13

(Prior Art)

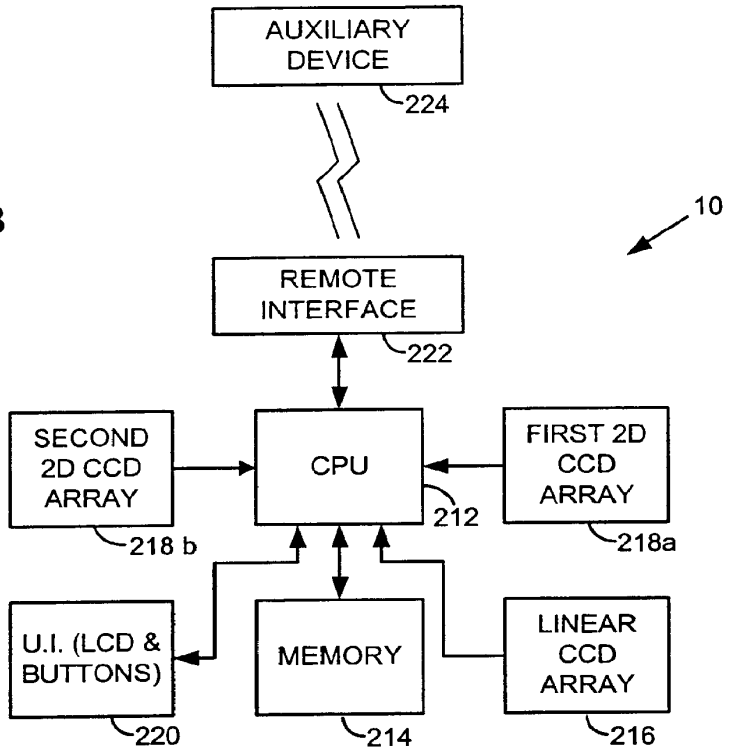
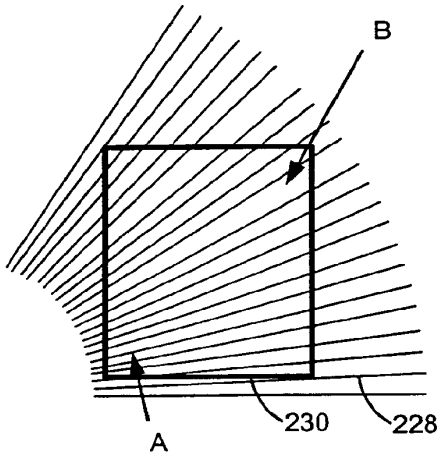


FIG. 14





Grid Orientation Under CCD 218a

Grid Orientation Under CCD 218b

FIG. 15

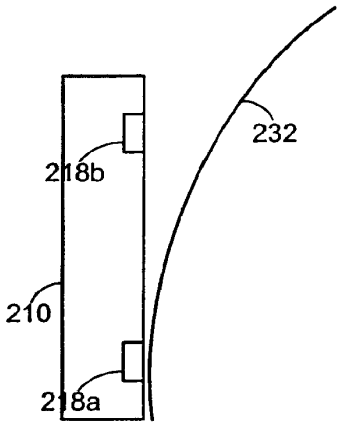


FIG. 16

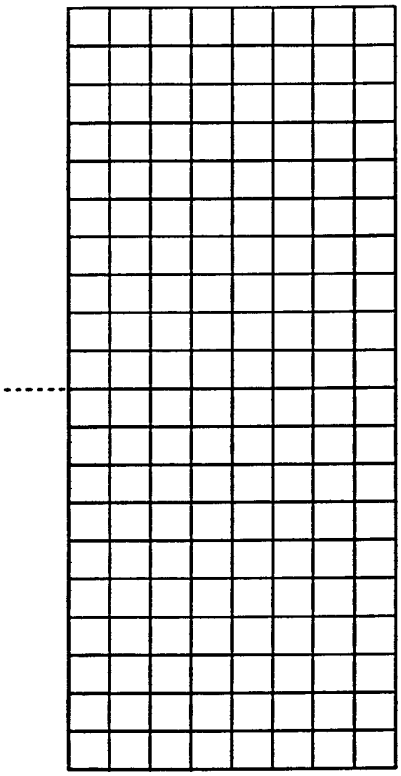


FIG. 17
(Prior Art)

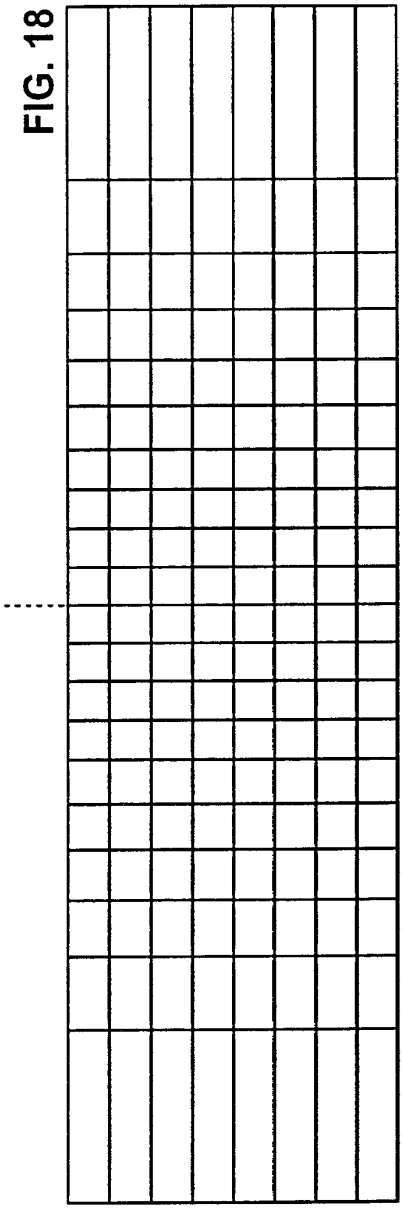


FIG. 18

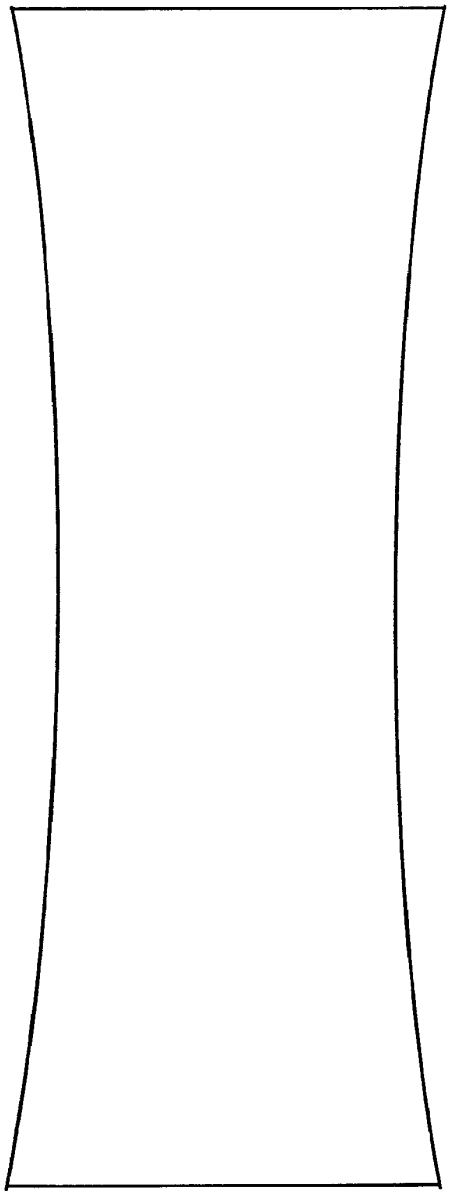


FIG. 19

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/13333

A. CLASSIFICATION OF SUBJECT MATTER
IPC(7) :G09C 5/00; G06F 17/00 ; G06K 7/00, 9/00, 9/36, 19/06; H04L 9/00
US CL :Please See Extra Sheet.
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : Please See Extra Sheet.

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,761,686 A (BLOOMBERG) 02 JUNE 1998, COL.1-24	1-36
Y, P	US 6,052,486 A (KNOWLTON ET AL.) 18 APRIL 2000, COL.2-62	1-36
A, P	US 5,930,767 A (REBER ET AL.) 27 JULY 1999, COL.1-11	1-36
A	US 5,841,886 A (RHOADS) 24 NOVEMBER 1998, COL.1-7	1-36
A, P	US 5,926,550 A (DAVIS) 20 JULY 1999, COL.2-5	1-36
A	US 5,862,260 A (RHOADS) 19 JANUARY 1999, COL.1-94	1-36

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search	Date of mailing of the international search report
23 AUGUST 2000	18 SEP 2000
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer GLEN BURGE James R. Matthews Telephone No. (703) 305-4792

Form PCT/ISA/210 (second sheet) (July 1998)*

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/13333

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,708,717 A (ALASIA) 13 JANUARY 1998, COL.2-12	1-36
A	US 5,168,147 A (BLOOMBERG) 01 DECEMBER 1992, COL.2-19	1-36
A	US 5,278,400 A (APPEL) 11 JANUARY 1994, COL.2-5	1-36

Form PCT/ISA/210 (continuation of second sheet) (July 1998)★

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/13333

Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This international report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. [X] Claims Nos.: 37
because they relate to subject matter not required to be searched by this Authority, namely:

It does not mean anything.

2. [ ] Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. [ ] Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. [ ] As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. [ ] As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. [ ] As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. [ ] No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

[ ] The additional search fees were accompanied by the applicant's protest.

[ ] No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/13333

A. CLASSIFICATION OF SUBJECT MATTER:
US CL :

235/454, 456, 456, 494; 283/17, 57, 58, 72, 73, 85, 93; 358/429, 455; 380/30, 51, 54, 55, 243;382/115, 232; 705/26, 77; 707/529; 713/176, 193

B. FIELDS SEARCHED
Minimum documentation searched
Classification System: U.S.

235/454, 456, 456, 494; 283/17, 57, 58, 72, 73, 85, 93; 358/429, 455; 380/30, 51, 54, 55, 243;382/115, 232; 705/26, 77; 707/529; 713/176, 193

PCT WORLD INTEL

INTERNATIONAL APPLICATION PUBLI

WO 9603286A1

(51) International Patent Classification 6 : B42D 15/10 — A1

(11) International Publication Number: **WO 96/03286**

(43) International Publication Date: 8 February 1996 (08.02.96)

(21) International Application Number: PCT/US95/09398

(22) International Filing Date: 26 July 1995 (26.07.95)

(30) Priority Data:
08/280,787 26 July 1994 (26.07.94) US

(71) Applicant: INTERNATIONAL DATA MATRIX, INC. [US/US]; Suite 200, 28100 U.S. 19 North, Clearwater, FL 34621-2656 (US).

(72) Inventor: PRIDDY, Dennis, G.; 806 Harbor Island, Clearwater, FL 34630 (US).

(74) Agent: ISACKSON, Robert, M.; Davis Hoxie Faithfull & Hapgood LLP, 45 Rockefeller Plaza, New York, NY 10111 (US).

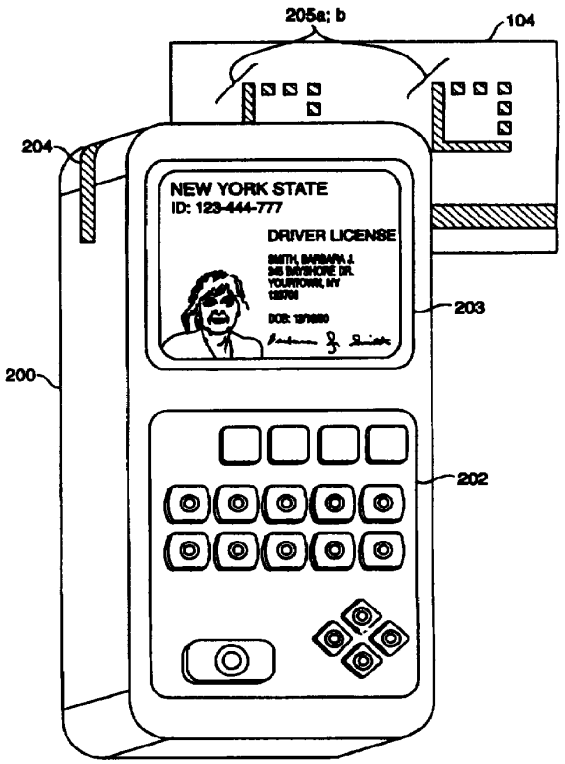
(81) Designated States: CA, CN, JP, SG, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(54) Title: UNALTERABLE SELF-VERIFYING ARTICLES

(57) Abstract

Disclosed are methods, systems and articles of manufacture for creating and authenticating self-verifying articles (104). Self-verifying articles include, for example, commercial instruments (i.e. notes, drafts, checks, bearer paper, etc.), transaction cards (i.e. ATM cards, calling cards, credit cards, etc.), personal identification documents (i.e., driver's licenses, passports, personal identification papers, etc.) and labels affixed to package surfaces for identification of the package owner or sender, which, for example, may be used for verifying imported goods by customs agents. Self-verifying article (104) creation includes receiving recipient specific data, encoding a first selected subset of the recipient specific data and fixing the encoded subset along with other human recognizable data on a surface of an article. Self-verifying article authentication includes scanning (502) a surface to locate an encoded first data set, decoding (503) the first data set and comparing (504) the decoded first data set with a control data set, which may also be fixed upon the surface, to determine (505) the authenticity of the received self-verifying article.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	**GB**	United Kingdom	**MR**	Mauritania
AU	Australia	**GE**	Georgia	**MW**	Malawi
BB	Barbados	**GN**	Guinea	**NE**	Niger
BE	Belgium	**GR**	Greece	**NL**	Netherlands
BF	Burkina Faso	**HU**	Hungary	**NO**	Norway
BG	Bulgaria	**IE**	Ireland	**NZ**	New Zealand
BJ	Benin	**IT**	Italy	**PL**	Poland
BR	Brazil	**JP**	Japan	**PT**	Portugal
BY	Belarus	**KE**	Kenya	**RO**	Romania
CA	Canada	**KG**	Kyrgystan	**RU**	Russian Federation
CF	Central African Republic	**KP**	Democratic People's Republic of Korea	**SD**	Sudan
CG	Congo			**SE**	Sweden
CH	Switzerland	**KR**	Republic of Korea	**SI**	Slovenia
CI	Côte d'Ivoire	**KZ**	Kazakhstan	**SK**	Slovakia
CM	Cameroon	**LI**	Liechtenstein	**SN**	Senegal
CN	China	**LK**	Sri Lanka	**TD**	Chad
CS	Czechoslovakia	**LU**	Luxembourg	**TG**	Togo
CZ	Czech Republic	**LV**	Latvia	**TJ**	Tajikistan
DE	Germany	**MC**	Monaco	**TT**	Trinidad and Tobago
DK	Denmark	**MD**	Republic of Moldova	**UA**	Ukraine
ES	Spain	**MG**	Madagascar	**US**	United States of America
FI	Finland	**ML**	Mali	**UZ**	Uzbekistan
FR	France	**MN**	Mongolia	**VN**	Viet Nam
GA	Gabon				

UNALTERABLE SELF-VERIFYING ARTICLES

TECHNICAL FIELD OF THE INVENTION

The present invention relates in general to encoding methods and systems, and in particular to methods, systems and articles of manufacture for creating and authenticating self-verifying articles.

BACKGROUND OF THE INVENTION

Modern life requires that individual identification and document authenticity be quickly, conveniently and reliably verified. The necessity for both individual and document verification arises in almost every commercial transaction. Additionally, the necessity for individual identification arises in both social and governmental settings with ever increasing frequency.

Commercial transactions requiring both document verification and individual identification include credit card, calling card, automatic teller machine ("ATM") and similar transactions, as well as other daily commercial transactions such as check cashing. For example, when a check is presented to a bank for payment, the bank is required to verify the authenticity of the check writer's signature (called the endorsement) and that there is enough money in the checking account to cover the check. Authenticity of the endorsement is determined by comparing the signature appearing upon the check with a signature sample of the check writer on file with the bank. A reasonably good forgery of the endorsement might enable an unauthorized person to illegally cash the check.

Alternatively, in non-commercial settings, identification issues often arise in the context of security. For example, common apartment and office building security systems require that anyone wishing to enter the building "sign in" in front of a security guard, and often, to present the guard with a previously issued

personal identification document which authorizes access to the building. The security guard is required to exercise his best personal judgment to determine that the identification document is authentic and that the person presenting it is the person identified on the identification document. In such circumstances, and understandably so, a security guard may be deceived by a person who is correctly identified on a forged or altered identification document. In the governmental context, it is a requirement in many countries that citizens carry personal identification papers in public to be produced for review upon request by appropriate authorities. For example, private individuals are required to present to the police personal identification, such as a driver's license, at the scene of a traffic accident or when stopped for a traffic violation. Additionally, personal identification documents are required for admittance to voting polls, and when crossing international boundaries and/or importing or exporting goods.

Accordingly, a pronounced need exists for unalterable self-verifying personal, commercial and governmental identification cards, papers, documents, labels, packaging and other similar articles. For the purposes of this patent document, an article shall be defined as any item having a surface, which may include a substrate, to which data may be fixed. As used herein, the term fixed shall mean one or more of the following, but is not limited to, attached, imprinted, adhered, etched, scratched, painted, printed, peened, embedded, machined, drilled, stamped, or otherwise imaged.

One current solution requires the use of biometric information that is stored in a memory device carried by an individual. The term biometric information refers to a characteristic personal to an individual, such as a signature, a finger print or a picture, for example. A sample of biometric information to be used is obtained from the person at an "encodation" site where the memory device is programmed under secure conditions. The sample is formed into a code by conventional encoding techniques. The sample may be obtained by having the person place a hand, eye, face or other unique physical feature upon a scanning input device. The scanned

information is then encoded to form a code which is subsequently stored to an alterable, portable memory device (i.e., magnetic strips, electronic or optical memory cards, floppy disks, etc.). The portable memory device is issued to the person. When the person's identity needs to be verified, the person presents the portable memory device at a "remote access/decodation" site where identification verification is to occur, and the information contained within the portable memory device is read from memory. Another sample of biometric information is then obtained by the person again placing a particular physical feature upon the input scanning device. The read code and the just sampled biometric information are compared by a machine to determine authenticity. In this regard, the read code may be decoded, e.g., using a process that reverses the encodation performed previously, or the sample information encoded, e.g., using the same encoding process used at the encryption site, to make the comparison. Because this method requires a processing system for performing data encoding and/or decoding, complex opto-electric hardware at every encodation and remote access site, and a memory device for each person, this solution is exceedingly expensive.

It is therefore an object of the present invention to provide an unalterable code, for use on an article, which contains biometric identification information personal to the authorized bearer of the article.

Another object of the present invention is to provide methods and systems for inexpensively, accurately and efficiently producing unalterable self-verifying personal and commercial articles.

A further object of the present invention is to provide methods and systems for accurately, efficiently and inexpensively authenticating presented self-verifying articles.

A still further object of the present invention is to provide methods and systems for verifying the authenticity of self-verifying articles presented at remote access sites which do not require expensive verification equipment, such as physical-trait-scanning input devices, or the inconvenience of a communication channel to a central location.

SUMMARY OF THE INVENTION

The invention is directed to a self-verifying article which contains an encoded machine-readable data set which includes recipient-specific biometric data. Self-verifying articles include, for example, commercial instruments (i.e., notes, drafts, checks, bearer paper, etc.), transaction cards (i.e., ATM cards, calling cards, credit cards, etc.), personal identification documents (i.e., driver's licenses, government benefit cards, passports, personal identification papers, etc.) and labels affixed to package surfaces, including for example identification of the package owner or sender, which may be used for verifying imported goods by customs agents. A subset of, or the whole of, the biometric data set may be, for example, a graphic image of a personal characteristic considered unique to a particular individual, such as, for example, a fingerprint, a retinal scan, a photo, a signature, etc., or some combination of the foregoing, is preferably encoded to generate a machine-readable data set. This article is preferably a low cost article, of paper or plastic, but may be any substrate, and the machine-readable data set is preferably fixed upon or in the article. The article also may, but need not, contain a human-readable version of the biometric data set.

In one embodiment of the invention, the encoded machine-readable data set is fixed upon the article in a manner which is neither comprehensible, nor detectable, by the human-eye, unless assisted by a suitably arranged reading device. For example, a check, or any other article for that matter, has fixed thereon a machine readable data set including the authorized user's signature. Thus, a would be forger would not have a sample of an authorized signature to copy. A comparison of the user's signature with the decoded signature permits verification at the site of use.

In another embodiment, a human-readable textual data set also appears on the article, optionally with selected subsets of the textual data set also being encoded and concatenated, interleaved, etc. with the encoded biometric data set. For the purpose of this

patent document, a textual data set comprises all data which is not biometric data.

One embodiment of the invention, therefore, is a self-verifying article which includes a surface, and a data set, the data set comprising an encoded recipient-specific biometric data in a machine readable form. Another embodiment is a self verifying article which includes a surface, a first data set and a second data set fixed upon the surface, the second data set being an encoded copy of the first data set. An alternate embodiment is a self-verifying recipient-specific identification article which includes a surface, a textual data set including at least one textual data subset fixed upon the surface, a biometric data set including at least one biometric data subset fixed upon the surface, and a machine-readable data set including an encoded copy of the textual data set and the biometric data set, which are optionally concatenated, interleaved or otherwise combined, and fixed upon the article. The machine-readable data set is preferably configured as an optically readable binary code forming at least one matrix (or array). The matrices are commonly referred to as two-dimensional bar codes or matrix codes.

One aspect of the invention concerns a method of producing an unalterable self-verifying article which includes the steps of receiving a recipient-specific data set comprised of one or more recipient-specific data subsets, generating a machine-readable data set by selectively encoding a first recipient-specific data subset, and fixing the machine-readable data set, and, optionally, the first recipient-specific data subset onto a surface of the article. The machine-readable data set may preferably be fixed in one or more machine-readable matrices. The imprinted machine-readable data set may be fixed as visual binary data, e.g., onto a designated blank area of the article, or alternately, as previously introduced, the machine-readable data set may be imprinted over or under already printed areas of the article utilizing marking means, such as an ink that can be detected separate from the article print, e.g., an ultraviolet, infrared, or other color ink, or by placing the machine-readable data set on the article so that a

permanent magnetic or fluorescent image may be selectively read. Yet another possibility, is that the machine-readable data set is fixed in a form that is discernable spectrally and the printing over the data set is a fake code that if photocopied or otherwise reproduced, produces an invalid code.

The machine readable data set alternatively may be physically fixed as binary data as voids. For the purposes of this patent document, voids shall include, but are not limited to, hollows, apertures, bubbles, detents, holes or the like or the absence thereof, which may optionally be configured in a matrix, such that the use of a physical material detection system capable of distinguishing the void/non-void areas is used. Such a detection system may include ultrasound, or other imaging technique, that has a return signal for determining the depth or density of a cell area to determine whether or not a void is present. Other optical techniques may be similarly used, such as are used in conventional compact disc technology. Preferably, when physical characteristics are used, a material may be used to fill in or cover any voids to provide the article with a smooth surface. Thus, the formation of an article may be a multilayer structure of which one layer contains the machine-readable code as voids.

Another alternative is an article having a layer upon which the machine-readable code is fixed that is opaque except for the code (or vice versa) so that a strong blacklight can be used to optically detect the code, not withstanding that the code is hidden by another layer.

Another aspect of the invention concerns a method of operation for verifying the authenticity of a self-verifying article. One such method includes scanning a self-verifying article of one of the aforementioned types which includes an encoded biometric data set, locating (or reading, etc.) and decoding the biometric data set, and comparing the decoded biometric data set to a recipient-specific sample and determining whether or not the decoded biometric data set corresponds to the recipient-specific sample. Another method includes the steps of receiving the self-verifying article, the self-verifying article including first and second data

sets wherein the first data set is an encoded copy of a second data set, scanning the self-verifying article to locate (or read, etc.) the encoded first data set, decoding the encoded first data set, and comparing the decoded first data set with the second data set to determine the authenticity of the self-verifying article. In the preferred embodiment, the encoded first data set has been imprinted in one or more machine-readable matrices.

A processing system, in accordance with the principles of the present invention, for producing a unique machine-readable data set for fixing upon an article, includes an input port operable to receive a recipient-specific data set which includes a plurality of recipient-specific data subsets, a memory storage device operable to store a plurality of processing system instructions, a processing unit for generating the machine-readable data set, and optionally an output port for transmitting the produced machine-readable data set and the first recipient-specific data subset. The processing unit retrieves and executes at least one of the processing system instructions from the memory storage device. The processing system instructions direct the processing unit to selectively encode the first recipient-specific data subset. In one embodiment of the invention, the processing unit is further operable to configure the machine-readable data set as an optically readable binary code forming at least one matrix.

A processing system, in accordance with the principles of the present invention, for verifying the authenticity of a self-verifying article, includes an input port operable to receive (or scan, read, etc.) the self-verifying article, a memory storage device operable to store a plurality of processing system instructions, a processing unit for verifying the authenticity of the self-verifying article, and optionally an output port operable to transmit an output signal. The input port includes means controlled by the processing unit, or alternatively by another processing unit or input control device, for selectively scanning the surface of the self-verifying article. The processing unit retrieves and executes at least one of the processing system instructions from the memory storage device which directs the

processing unit to locate an encoded first data set fixed upon the self-verifying article and to decode the encoded first data set. In one embodiment, the processing unit will further operate to compare the decoded first data set with a second data set fixed to the self-verifying article, and to generate an output signal indicating the authenticity of the self-verifying article. In an alternate embodiment, the processing unit will transmit the decoded first data set and the second data set to a central host processing system operable to compare the two data sets. In another embodiment, the decoded first data set will include biometric data which the output port will transmit to an output display device enabling an attendant to perform a visual comparison for identification verification of the bearer of the self-verifying article.

One embodiment for using and/or distributing the present invention is as software stored to a storage medium. The software includes a plurality of computer instructions for controlling one or more processing units for producing and/or authenticating unique self-verifying articles in accordance with the principles of the present invention. The computer will include the necessary encodation and/or decodation procedure/algorithms, or parts thereof, to be used. The storage mediums utilized may include, but are not limited to, magnetic storage, optical memory, and/or semiconductor chip, to name three examples.

Accordingly, an advantage of the present invention is the ability to provide an unalterable code for use on an article, the article containing biometric identification and character information personal to the authorized bearer of the article.

Another advantage is that the machine-readable code is fixed on the article using relatively inexpensive techniques, preferably by conventional printing equipment, that are highly reliable and accurate.

Another advantage of the present invention is the ability to provide self-verifying articles, as well as methods and systems for inexpensively, accurately and efficiently producing unalterable

self-verifying personal identification documents and commercial instruments.

A further advantage of the invention is the ability to provide methods and systems for accurately, efficiently and inexpensively authenticating a presented self-verifying article.

A still further advantage of the invention is the ability to provide methods and systems for verifying the authenticity of self-verifying articles presented at remote access sites without requiring expensive verification equipment.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention, and the advantages thereof, reference is now made to the following descriptions taken in conjunction with the accompanying drawings in which like numbers designate like parts, and in which:

FIG. 1A illustrates a functional block diagram of a system for producing a self-verifying article in accordance with the principles of the present invention;

FIG. 1B illustrates an isometric view of the processing system set forth in FIG. 1A;

FIG. 1C illustrates a block diagram of a processing unit and a memory storage device;

FIG. 2A illustrates a functional block diagram of a system for verifying the authenticity of a received self-verifying article in accordance with the principles of the present invention;

FIG. 2B illustrates an isometric view of the remote access site processing system set forth in FIG. 2A;

FIG. 3 illustrates a machine-readable binary coded matrix;

FIGS. 4A and 4B illustrate flow diagrams for producing an unalterable self-verifying article in accordance with the embodiment illustrated in FIG. 1A; and

FIG. 5 illustrates a flow diagram for verifying the authenticity of a received self-verifying article in accordance with the embodiment illustrated in FIG. 2A.

DETAILED DESCRIPTION OF THE INVENTION

FIG. 1A illustrates a functional diagram of a system for producing a self-verifying article in accordance with the present invention. The system includes an input data set 100, which includes a biometric data set 101 and an optional textual data set 102, an article 10, a processing system 103 and a self-verifying article 104. As previously introduced, data set 100 is comprised of recipient specific data. Biometric data set 101 may include one or more physical traits personal to the potential article recipient (i.e., photo, retinal scan, finger print, signature, etc.), while textual data set 102, which is optionally included in input data set 100, may include one or more textual attributes (i.e., name, address, height/weight, eye color, etc.). Processing system 103 is operable to produce self-verifying article 104 by generating a unique machine-readable data set for fixing upon article 10.

Processing system 103 includes input means, processing means, output means and article producing means. The input means are for receiving input data set 100 and article 10. The processing means are for validating input data set 100, both syntactically and semantically, and encoding selected subsets of biometric data set 101, and, optionally, selected portions of textual data set 102 (optionally, the data to be encoded may first be encrypted, if desired). The output means are for transmitting a validated and encoded data set, along with the selected subsets of biometric data set 101, and optionally with the textual data set 102, to the article producing means. The article producing means are for fixing the validated and encoded data set, and for optionally fixing the selected subsets of biometric data set 101 and optionally textual data set 102, to article 10, thereby producing self-verifying article 104.

In the preferred embodiment, processing system 103 ensures data integrity by encoding all selected biometric and textual data subsets into a compact unalterable machine-readable data set, and subsequently, configuring the machine-readable data set as one or more matrices. If preferred, the machine-readable data set may be divided into two or more individual segments, which segments may

then be incorporated into two or more two-dimensional machine-readable matrices, which may or may not appear visually similar in size. The multiple matrices, though physically separate, may contain check values and features which assure the detection of any attempted alteration of either the human-readable text and/or the machine-readable matrices. In this regard, the encoded biometric data and textual data can be concatenated into one data string, divided by approximately two, and then formed into two matrices of approximately equal size. Alternatively, the biometric data and text data may be interleaved, for example, in alternating bits, bytes, group of bytes, etc. to form a data string which is then divided into the two matrices. Preferably, each matrix is provided with a checksum for verifying the data integrity of each matrix independently. In addition, or alternatively, the matrices may have an interdependent checksum that is used to verify the data integrity of both matrices collectively. As a result of these checksums, if one matrix is altered, or if both matrices are altered, invalid data will be read. Advantageously, interleaving biometric data and textual data according to a predetermined routine enhances the ability to detect altered matrices. Alternatively, biometric data set 101 could be formed as one matrix and textual data set 102 could be formed as a second matrix.

In one embodiment, enhanced data security can be obtained and maintained by verifying the machine-readable data set for acceptability against predetermined criteria which may include searching a data base (e.g., an organized, comprehensive collection of data stored for use by processing system(s)) of previously issued articles to determine uniqueness. Note, in an alternate embodiment input data set 100 may be received at a remote access encodation site not equipped with the verification and/or encodation algorithm(s), in which case, a data signal representation of both the recipient-specific biometric data and the related textual data may be transmitted to a secure central host (shown in a similar context in FIG. 2A), the central host then performs the above mentioned verification. The transmission may be by wired or non-wired communication.

If the recipient is determined acceptable, the recipient-specific biometric data is encoded, which preferably includes utilization of compression algorithms, combining the biometric data subsets, and optionally, the textual data subsets, into one or more machine-readable matrices. If input data set 100 was received at a remote access encodation site not equipped with the encodation algorithm(s), the resulting encoded binary string is transmitted as previously discussed to the remote access encodation site. A standard article issuance device (shown in FIG. 1B) then fixes the machine-readable data set onto one or more self-verifying articles. As the articles are created and ejected from the article issuance device, a record of the event is automatically entered into the data base, which, if input data set 100 is received at a remote access encodation site, may be located at the central host. Entry of the record insures that duplicate articles are not inadvertently issued at a later date. Note that the number of articles issued is directly related to the intended use of the articles. Applications of this aspect of the invention include issuing only a single driver's license having a unique encoded photo, or issuing multiple articles, such as checks, traveler's checks, bank account withdrawal slips, etc. having the same encrypted signatures.

FIG. 1B illustrates an isometric view of processing system 103. Processing system 103 includes a personal computer ("PC") 105 coupled with a article issuance device 114. PC 105 is comprised of a hardware casing 106 (shown as having a cut away view), a monitor 109, a keyboard 110 and optionally a mouse 113. Hardware casing 106 includes both a floppy disk drive 107 and a hard disk drive 108. Floppy disk drive 107 is operable to receive, read and write to external disks, while hard disk drive 108 is operable to provide fast access data storage and retrieval. Although only floppy disk drive 107 is illustrated, PC 105 may be equipped with any suitably arranged structure for receiving and transmitting data, including, for example, tape and compact disc drives, and serial and parallel data ports. Within the cut away portion of hardware casing 106 is a processing unit, central processing unit ("CPU") 111, coupled

with a memory storage device, which in the illustrated embodimer is a random access memory ("RAM") 112. Although PC 105 is show having a single CPU 111, PC 105 may be equipped with a plurality CPUs 111 operable to cooperatively carry out the principles of the present invention. Article generating device 114 is operable to receive one or more output data sets from PC 105, and fix the output data sets to the article's surface.

Although PC 105 and article generating device 114 have been utilized for illustrating one implementation of processing system 103, the invention may alternately be implemented within any processing system having at least one processing unit, including, for example, sophisticated calculators and hand held, mini, main frame and super computers, including RISC and parallel processing architectures, as well as within network combinations of the foregoing, and may utilize any suitably arranged article producing means.

FIG. 1C illustrates a conceptual block diagram of one of any number of sub-processing systems which may be utilized in conjunction with FIGS. 1A and 1B. The sub-processing system includes a single processing unit, such as CPU 111, coupled via data bus 118 with a memory storage device, such as RAM 112. Memory storage device 112 is operable to store one or more instructions which processing unit 111 is operable to retrieve, interpret and execute. Processing unit 111 includes a control unit 115, an arithmetic logic unit ("ALU") 116, and a local memory storage device 117, such as, for example, stackable cache or a plurality of registers. Control unit 115 is operable to fetch instructions from memory storage device 112. ALU 116 is operable to perform a plurality of operations, including addition and Boolean AND needed to carry out instructions. Local memory storage device 117 is operable to provide high speed storage used for storing temporary results and control information.

FIG. 2A illustrates a functional block diagram of a system for verifying the authenticity of a received self-verifying article in accordance with the present invention. The system includes self-verifying article 104, a remote access site processing system 200,

optionally coupled with a central host processing system 103 (as indicated by the broken line), and an authenticity message displaying means 201, such as a display device, a printer, or other suitably arranged indicating device. Self-verifying article 104 includes at least one encoded data set which includes a first data subset that is an encoded copy of a portion, or the whole, of a biometric data set. Self-verifying article 104 also preferably includes a text data set or a biometric data set, or both.

Remote access site processing system 200 includes input means, processing means and output means. The input means are for receiving self-verifying article 104. The processing means are for verifying the authenticity of self-verifying article 104, which may include communications between remote access site processing system 200 and central host processing system 103. The output means are for transmitting an authenticity message produced by the processing means to display means 201.

The processing means are operable to scan self-verifying article 104 to locate and decode the encoded first data set and to compare the decoded first data set with a second data set, which is either obtained from the bearer of the article or is fixed to self-verifying article 104, and to generate the output signal indicating authenticity of article 104. In an alternate embodiment, the processing means either selectively bypasses or is not operable to perform the comparison of the decoded first data set and the second data set. Instead, the processing means generates an output signal representative of the decoded first data set, e.g., a graphic image display of the portion of the biometric data set, and the second data set to a display device for manual comparison and verification by a processing system operator. Alternatively, the processing system operator may manually compare the decoded first data set, and optionally the second data set (if one is fixed to the article), with the article bearer or a biometric data set obtained from the bearer, e.g., the bearer's signature or appearance, or from a database.

The illustrated system for verifying the authenticity of self-verifying article 104 may utilize a variety of devices including,

for example, portable terminals, fixed station readers, and flat bed scanners, each of which may directly incorporate decoder capability or have decoder capability available at a base/host station, such as processing system 103, via wired or radio frequency, short wave, cellular, infrared or other form of non-wired communication. Remote access site processing system 200 and/or central host processing system 103 may be configured with keyboards and display screens of sufficient resolution to accurately display the encoded biometric image and/or textual data, and may incorporate imaging apparatus necessary to convert machine-readable data sets into binary machine language bits in preparation for decoding. The imaging apparatus may be based on any of a number of technologies, including CCD, CMOS, and NMOS or other forms of light sensitive sensors, which sensors may be structured in the form of a two-dimensional area or one-dimensional linear arrays, or a single beam laser reading for scanning a two-dimensional image in a raster pattern.

One preferred embodiment of imaging apparatus is a linear array scanner that is vertically aligned with a solid border of an imprinted machine readable code 205, and when two or more matrices are used, these matrices are arrayed in parallel orientation so that the two symbols can be passed by the CCD scanner using a conventional card swipe action as in conventional magnetic strip reading. The matrices are then read and a video image of each matrix is stored in memory for processing. Imaging may also be achieved through the use of lasers, laser-diodes, infrared or other such binary imaging technologies which devices may also be structured in the form of two-dimensional area or one-dimensional linear arrays. Additionally, readers may include the ability to automatically verify the images and information encoded within the machine-readable matrix to the human recognizable version on the same article. In one embodiment, this comparison may be accomplished internal to remote access site processing system 200's memory thereby precluding the need for key pads and/or high resolution display screens on the terminals. Alternately, as introduced, operators may visually compare the information

displayed on the terminal screen to the human readable information now present on the article and/or to the article bearer.

FIG. 2B illustrates an isometric view of a hand held computer which may be used as remote access site processing system 200. Hand held computer 200 includes a keypad 202, a display screen 203 and an input port 204. Keypad 202 includes a systematic arrangement of keys for manually receiving input data from a user. Display screen 203 is for displaying an authenticity message, and/or biometric and/or textual data. Input port 204 is for receiving self-verifying article 104, here illustrated as a driver's license, which in the illustrated embodiment includes encrypted machine-readable data sets 205a;b configured as two optically readable binary matrices. Remote access site processing system 200 includes at least one processing unit and one memory storage device, such as the sub-processing system illustrated in FIG. 1C. Preferably, the processing unit includes a microprocessor having associated memory (non-volatile storage for containing the program instruction set to identify and decode the matrices, and volatile memory for a data processing work area), a video memory for storing an image of the matrices to be decoded, and associated signal conditioning circuits, which are mounted on a single printed circuit board.

FIG. 3 illustrates a preferred single machine-readable binary coded matrix, generally indicated as a matrix 205. Matrix 205 is a sample of the Data Matrix symbology developed by International Data Matrix, Inc., Clearwater, Florida, the assignee of this invention. Matrix 205 has a perimeter 300 formed by intersecting sides 301 formed of solid lines and intersecting perimeter sides 302 formed of dark perimeter squares 303 and light perimeter squares 304 in an alternating pattern. Data, generally indicated as 305, is stored within perimeter 301 of matrix 204 by converting each character to be stored to a visual binary code represented by dark and light squares corresponding to ones and zeros of encoded binary information. For a more complete description of the configuration of matrix 205, reference is made to United States Patent No. 4,939,354, entitled "Dynamically Variable Machine

Readable Binary Code and Method for Reading and Producing Thereof" and to co-pending patent application, United States patent 5,324,923, entitled "Apparatus for Producing a Dynamically Variable Machine Readable Binary Code and Method for Reading and Producing Thereof," commonly owned by the assignee of this patent document, and which are incorporated herein by reference.

FIG. 4A illustrates a flow diagram for producing an unalterable self-verifying article in accordance with the embodiment illustrated in FIG. 1A. Upon entering START block 400, the process according to the principles of the present invention begins. The recipient-specific data set, which is comprised of at least one data subset, is received by processing system 103 (input block 401). Processing system 103 preferably performs a graphic image compression of a first data subset. Image compression by a factor of approximately 50:1 or better is preferred to obtain a digital representation of the acquired data. Such compressed data is capable of reproducing on a conventional graphic display screen the recipient specific image without any significant degradation of visual quality (block 402). The image compression may be by any standard routine e.g., Discrete Cosine Transform (DCT), LZW (Lempel-Ziv), fractal, or others to reduce the amount of bits required to encode the first data subset. A compression ratio of 50:1 is deemed suitable, but other ratios may be used. In addition to data compression, a graphic image enhancement routine may be performed on the first data subset, preferably before the data compression step, in order to enhance the image contrast, sharpen and smooth edges and reduce the effect of shadows, particularly for imaging a photograph of the recipient. The foregoing improves the digital image for more effective data compression. Suitable image enhancement routines are known, and described R. Gonzalez et al., Digital Image Processing, published by Addison-Wesley Publishing Co. (Reading MA) 1987. Processing system 103 selectively encodes the compressed first data set, thereby generating a machine-readable data set (processing block 403). This selectively encoding step is more fully discussed in connection with the detail description of FIG. 4B. In one embodiment of the invention,

processing system 103 is further operable to configure the machine-readable data set as an optically readable binary code forming one or more matrices (processing block 404). Processing system 103 fixes the machine-readable data set and the first recipient-specific data subset onto a surface of an article, thereby producing self-verifying article 104 (processing block 405). In one embodiment, the matrix is fixed on the article using a conventional printing process, e.g., thermal, thermal transfer, ink-jet, bubble-jet, laser jet, dot matrix printing, etc. Alternatively, the matrix or matrices may be fixed sub-surface, e.g., by laminating a top surface or by placing the matrix over the printed layer of a multilayer article. In another embodiment, the machine-readable data set is imprinted over an already printed area of the article, such as the photograph on a driver's license for example. In yet another embodiment, the matrix is formed by introducing bubbles or voids into the article, or drilling or punching holes into or through the article, according to the matrix pattern, such that the code is machine readable by a technique capable of detecting the absence or presence of material, or the relative density of material, or the depth of a bubble, void, hole or the like in the article, e.g., ultrasonically, or by a light measuring system or other suitable imaging system having a bounceback signal capable of distinguishing the code.

FIG. 4B illustrates a more detailed flow diagram of processing block 402 illustrated in FIG. 4A. Upon entering START block 406, the selective encodation of the first data set begins. Processing system 103 compares the first data set with system control values to determine if the first data set is within acceptable tolerances (processing block 407). The step of comparing may include, for example, syntactic and/or semantic analysis. If it is determined that the first data set is invalid as received ("N" branch of decisional block 408), then processing system 103 aborts self-verifying article production (termination block 409). Alternatively, if it is determined that the first data set is valid as received ("Y" branch of decisional block 408), then processing system 103 searches a data base of previously issued articles to

determine if the issued article is unique (processing block 410), uniqueness being determined on a subjective basis as a function of the type of article being produced. Note that the data base utilized by processing system 103 may be internal or external to processing system 103, and that in either system, processing system 103 may search the data base directly, or indirectly. For example, the data base may be stored remotely and controlled by another processing system with which processing system 103 communicates. If it is determined that the first data set is not unique as received ("N" branch of decisional block 411), then processing system 103 aborts self-verifying article production (termination block 412). Alternatively, if it is determined that the first data set is valid as received ("Y" branch of decisional block 411), then processing system 103 selectively inserts one or more subsets of the received recipient-specific data set as at least one record into the data base (processing block 413). Processing system 103 then encodes the first data set (processing block 414), and in one embodiment adds error correction bits to the encoded first data set (processing block 415).

The selective encodation of only the first data set as embodied within FIGS. 4A and 4B was for illustrative purposes only, and it is understood that among the various aspects and features of the present invention is the ability to selectively encode a plurality of compressed recipient-specific data subsets, and to subsequently concatenate, interleave, etc. the encoded subsets, thereby forming a single machine-readable data set. Further, when two or more data subsets are encoded and concatenated, interleaved, etc. together, processing system 103 is operable to configure the machine-readable data set into one or more optically readable matrices wherein individual encoded data subsets may span two or more matrices.

FIG. 5 illustrates a flow diagram for verifying the authenticity of a received self-verifying article in accordance with the embodiment illustrated in FIG. 2A. Upon entering START block 500, the process according to the principles of the present invention begins. The self-verifying article, which in this

embodiment includes a plurality of data sets wherein a first data set is an encoded copy of a second data set, is received by a remote access site processing system 200 (input block 501). Remote access site processing system 200 then scans the received self-verifying article to locate the encoded first data set (processing block 502). Remote access site processing system 200 decodes the encoded first data set (processing block 503), and compares the decoded first data set with the second data set to determine the authenticity of the received self-verifying article (processing block 504).

In one embodiment, the comparison step is accomplished through communications between remote access site processing system 200 and processing system 103, wherein processing system 103 maintains a data base of recipient-specific data relating to previously issued self-verifying articles. In this embodiment, communications between remote access site processing system 200 and processing system 103 may be accomplished via wired or non-wired communication means. In an alternate embodiment, at least the decoded first data set, and optionally the second data set, are transmitted to an output display device for manual comparison by a system operator. If it is determined that the decoded first data set is not authentic ("N" branch of decisional block 505), then remote access site processing system 200 displays an authenticity message 201 indicating that the self-verifying article is invalid (output block 506). Alternatively, if it is determined that the decoded first data set is authentic ("Y" branch of decisional block 505), then remote access site processing system 200 displays an authenticity message 201 indicating that the self-verifying article is valid (output block 507).

In another embodiment, prior to decoding the encoded first data set, remote access site processing system 200 converts the received self-verifying article into a digital bit-map image, and separates the digital bit-map image into a plurality of regions, wherein a first region includes the encoded first data set and a second region includes the second data set. In this embodiment, both the first and second regions may include a plurality of

biometric and/or textual data subsets which remote site processir system 200 is further operable to convert into common data format for processing.

As noted previously, one embodiment of the self verifying article contains two matrices which have a first data set of biometric data and a second data set of textual data. In addition, in one embodiment, the article also may contain a magnetic stripe for containing alterable data, which may be programmed by scanning the machine readable matrices, decoding certain data contained therein and, decoding that data (with or without other data), onto the magnetic stripe. This makes the self verifying article useful in applications which require reading a magnetic stripe.

A further use of the invention is to prevent software piracy. Software is a special form of program which has been recorded to a storage medium, such as one of the above identified storage mediums. Software enables programs to be freely transferred or copied from one storage medium to another, which enables unlicensed users to obtain illegal copies of the software. For example, in one embodiment, the purchaser of a processing system provides a hardware vendor with industry standardized personal data, which may include biometric data, which is, optionally encrypted, and stored internal to the processing system. Whenever the processing system purchaser buys software, the purchaser is again required to provide such industry standardized personal data, which is compressed, optionally encrypted, encoded into a machine-readable data set, preferably as one or more binary coded matrices, and fixed to the surface of a portable storage medium, such as a floppy or compact disk. When the software is loaded onto the processing system, the matrices are scanned, decoded and verified in accordance with the principles of the present invention, compared with the previously stored data to ensure a commonality of ownership, thereby limiting software piracy. If common ownership is found, then the software is loaded onto the processing system along with the decoded industry standardized personal data. In the event the processing system owner transfers ownership of the processing system, the new owner, in order to load his software, will have to have the

industry standardized personal data redefined, which could suspend the new owner's use of the existing software or automatically delete the existing software. In the event use of the software is suspended, a "transfer" of ownership routine might be available to reactivate the suspended use of the existing software if ownership of the particular software was legally transferred.

Although the present invention and its advantages have been described in detail, it should be understood that various changes, substitutions and alterations can be made herein without departing from the spirit and scope of the invention.

WHAT IS CLAIMED IS:

1. A self-verifying article comprising:
a surface;
a first data set fixed upon said surface; and
a second data set fixed upon said surface, said second data set being an encoded copy of at least a portion of said first data set.

2. The self-verifying article as set forth in Claim 1 wherein said second data set comprises biometric data.

3. The self-verifying article as set forth in claim 1 wherein portions of said first and second data sets comprise biometric data.

4. The self-verifying article as set forth in Claim 1 wherein said second data set is configured as an optically readable binary code forming at least one matrix array.

5. The self-verifying article as set forth in Claim 2 wherein said second data set is configured as an optically readable binary code forming two matrix arrays.

6. The self-verifying article as set forth in Claim 4 further comprising:
a third data set fixed upon said article, said third data set comprising textual data.

7. The self-verifying article as set forth in Claim 6 further comprising:
a fourth data set fixed upon said article, said fourth data set is an encoded copy of at least a portion of said third data set.

8. The self-verifying article as set forth in Claim 7 wherein said second and fourth encoded data sets are combined together forming one machine-readable data set.

9. The self-verifying article as set forth in Claim 1 wherein said second data set is fixed upon said article and is undetectable to the human-eye.

10. The self-verifying article as set forth in Claim 1 wherein said article is a commercial instrument and said second data set includes an endorsement of the drawer/maker/owner of said commercial instrument.

11. The self-verifying article as set forth in Claim 1 wherein said article is a transaction card.

12. The self-verifying article as set forth in Claim 1 wherein the article further comprises a first outer surface and a second outer surface wherein the second data set is formed as an array of voids between the outer surfaces of said article.

13. A recipient-specific identification article comprising:

a surface; and

a machine-readable data set fixed upon said surface, said machine-readable data set including encoded biometric data configured as an optically readable binary code forming at least one matrix.

14. The recipient-specific identification article as set forth in Claim 13 wherein said fixed machine-readable data set is not detectable by an unaided human-eye.

15. The recipient-specific identification article as set forth in Claim 13 wherein said article is a transaction card.

16. The recipient-specific identification article as set forth in Claim 13 wherein said article is a commercial instrument.

17. The recipient-specific identification article as set forth in Claim 13 further comprising a textual data set fixed upon said surface, wherein said machine-readable data set includes an encoded subset of said textual data set.

18. The recipient-specific identification article as set forth in Claim 17 wherein said machine-readable data set is configured as an optically readable binary code forming two matrices.

19. The recipient-specific identification article as set forth in Claim 18 wherein said encoded biometric data and said encoded subset of textual data are combined to form a single data set.

20. The recipient-specific identification article as set forth in Claim 19 wherein said encoded biometric data and said encoded subset of textual data are concatenated together to form said single data set.

21. The recipient-specific identification article as set forth in Claim 19 wherein said encoded biometric data and said encoded subset of textual data are interleaved to form said single data set.

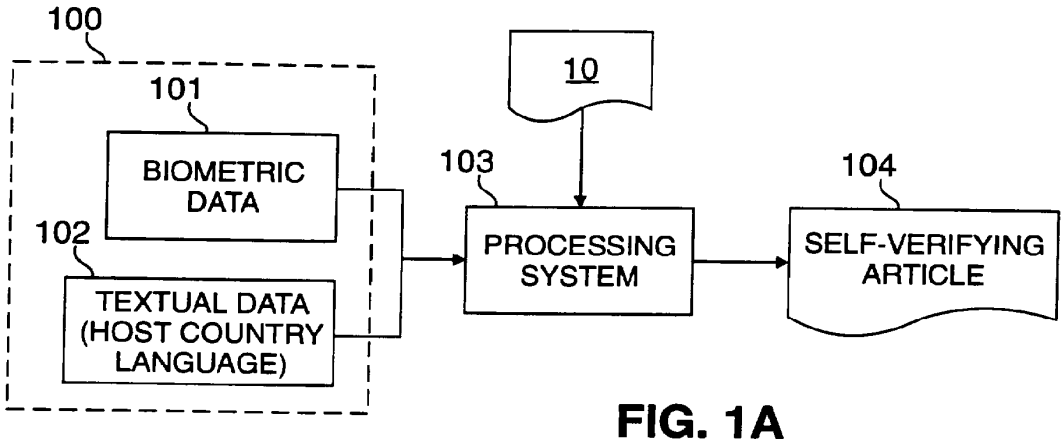

100
101
BIOMETRIC DATA
102
TEXTUAL DATA (HOST COUNTRY LANGUAGE)
10
103
PROCESSING SYSTEM
104
SELF-VERIFYING ARTICLE

FIG. 1A

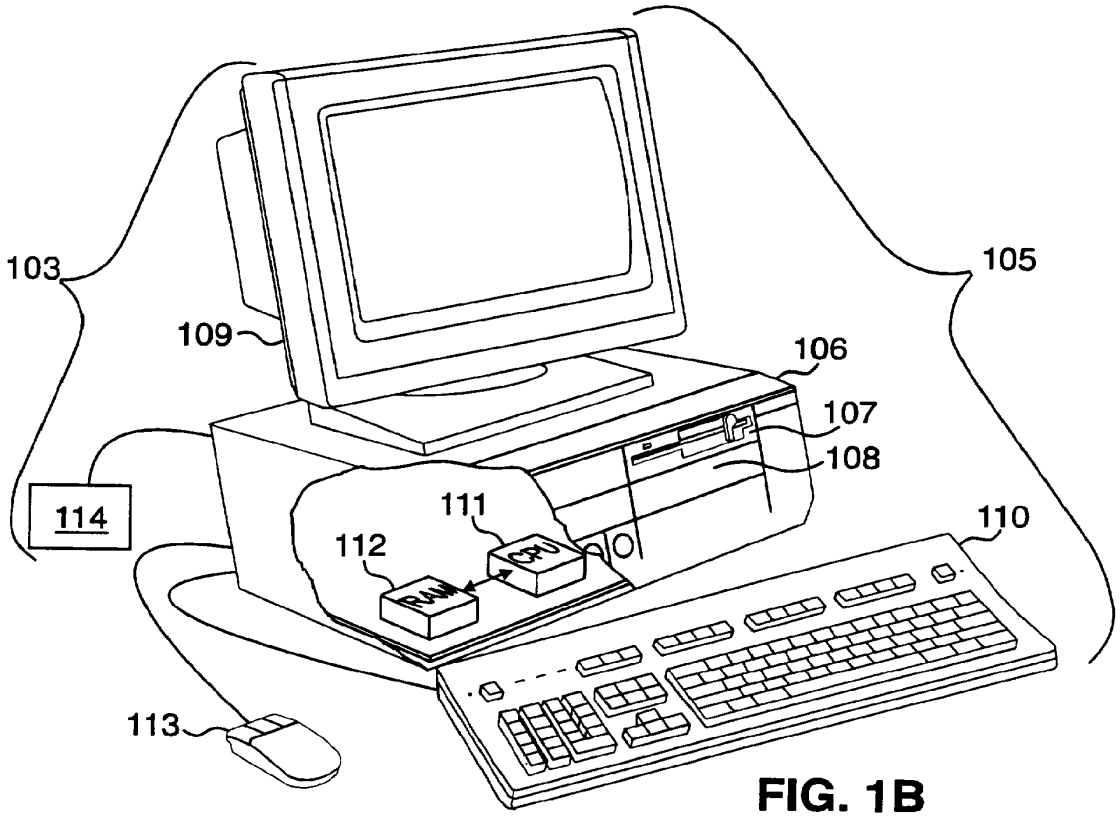

103
105
109
106
107
108
114
111
112
CPU
RAM
110
113

FIG. 1B

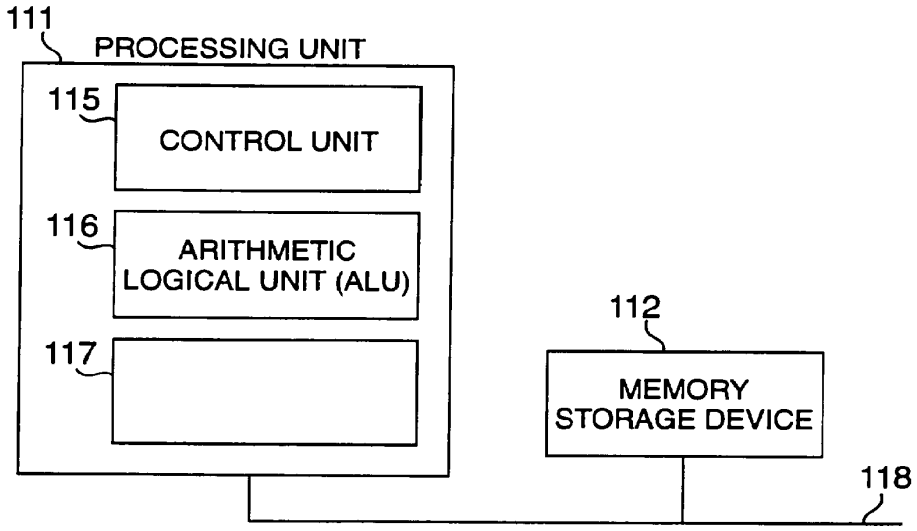

111
PROCESSING UNIT
115
CONTROL UNIT
116
ARITHMETIC LOGICAL UNIT (ALU)
117
112
MEMORY STORAGE DEVICE
118

FIG. 1C

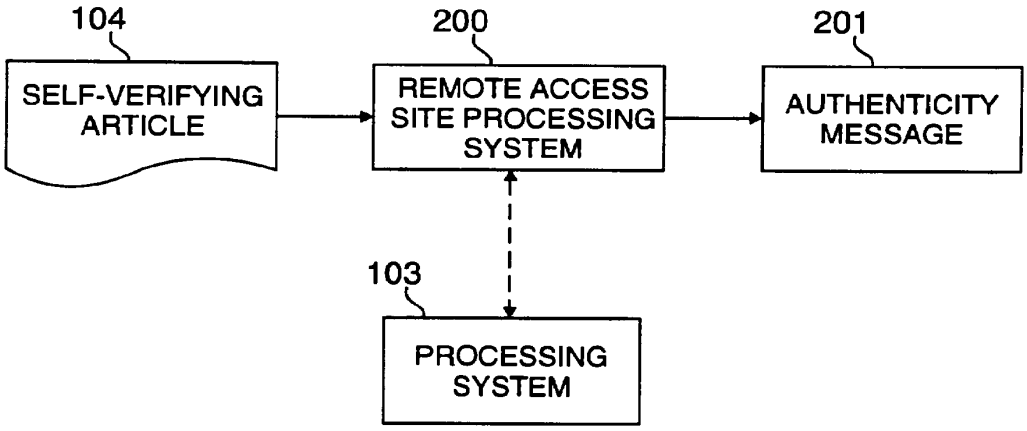

104
SELF-VERIFYING ARTICLE
200
REMOTE ACCESS SITE PROCESSING SYSTEM
201
AUTHENTICITY MESSAGE
103
PROCESSING SYSTEM

FIG. 2A

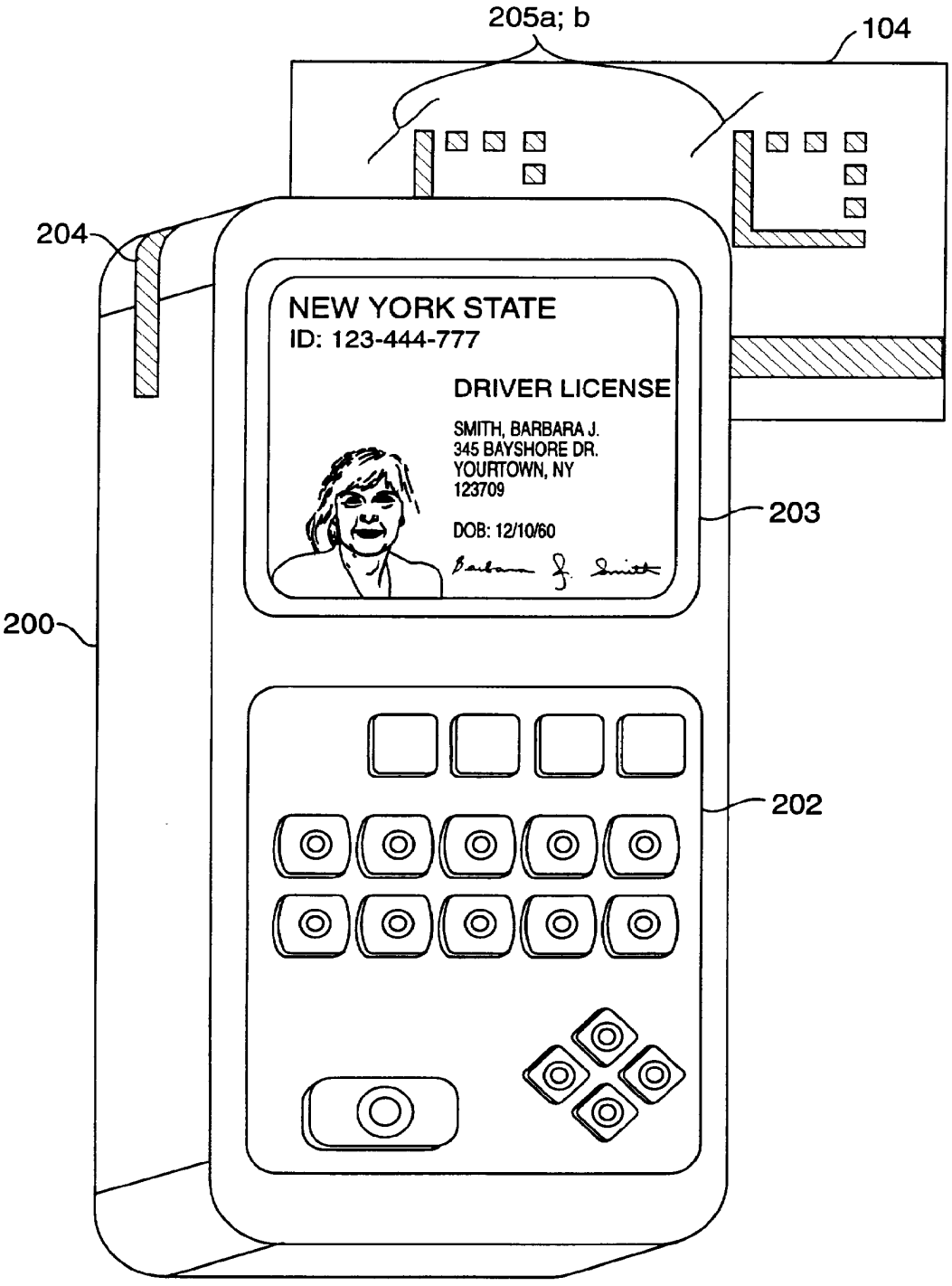

205a; b
104
204
NEW YORK STATE
ID: 123-444-777
DRIVER LICENSE
SMITH, BARBARA J.
345 BAYSHORE DR.
YOURTOWN, NY
123709
DOB: 12/10/60
203
200
202

FIG. 2B

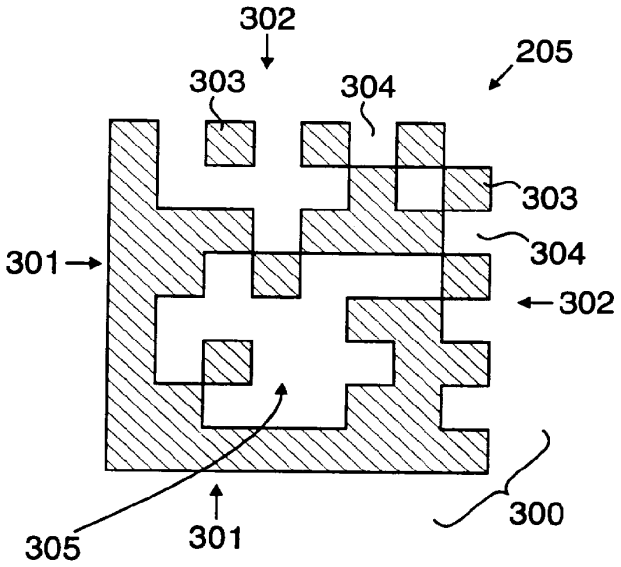

302
205
303
304
303
304
301
302
305
301
300

FIG. 3

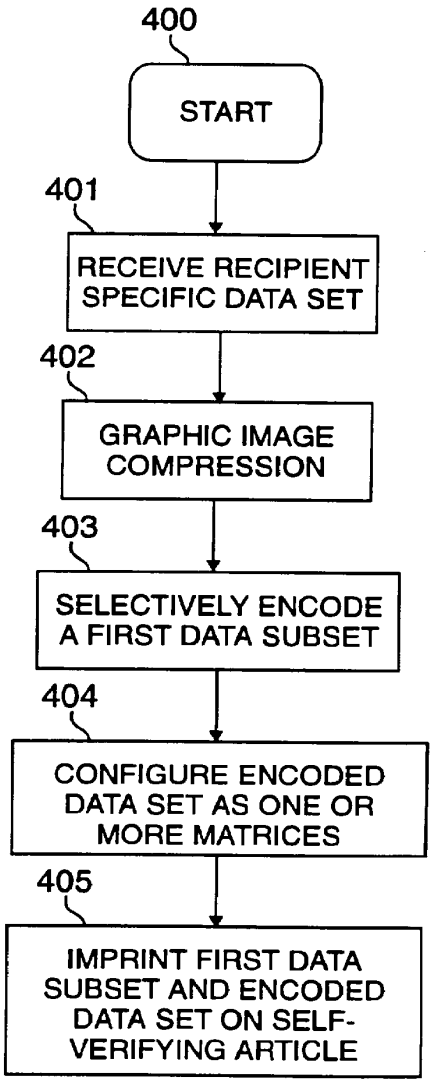

400
START
401
RECEIVE RECIPIENT SPECIFIC DATA SET
402
GRAPHIC IMAGE COMPRESSION
403
SELECTIVELY ENCODE A FIRST DATA SUBSET
404
CONFIGURE ENCODED DATA SET AS ONE OR MORE MATRICES
405
IMPRINT FIRST DATA SUBSET AND ENCODED DATA SET ON SELF-VERIFYING ARTICLE

FIG. 4A

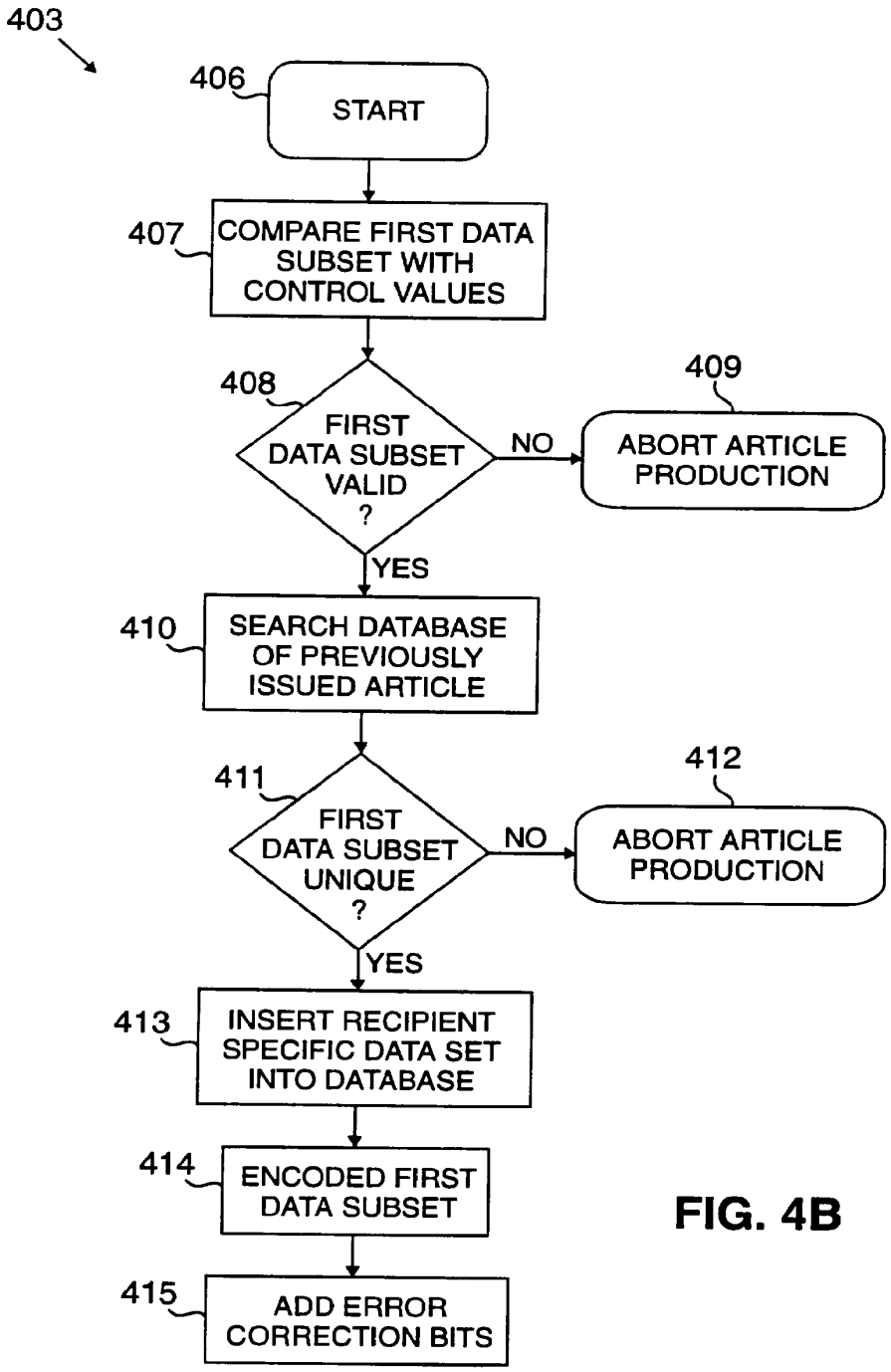

403
406
START
407
COMPARE FIRST DATA SUBSET WITH CONTROL VALUES
408
FIRST DATA SUBSET VALID ?
NO
409
ABORT ARTICLE PRODUCTION
YES
410
SEARCH DATABASE OF PREVIOUSLY ISSUED ARTICLE
411
FIRST DATA SUBSET UNIQUE ?
NO
412
ABORT ARTICLE PRODUCTION
YES
413
INSERT RECIPIENT SPECIFIC DATA SET INTO DATABASE
414
ENCODED FIRST DATA SUBSET
415
ADD ERROR CORRECTION BITS

FIG. 4B

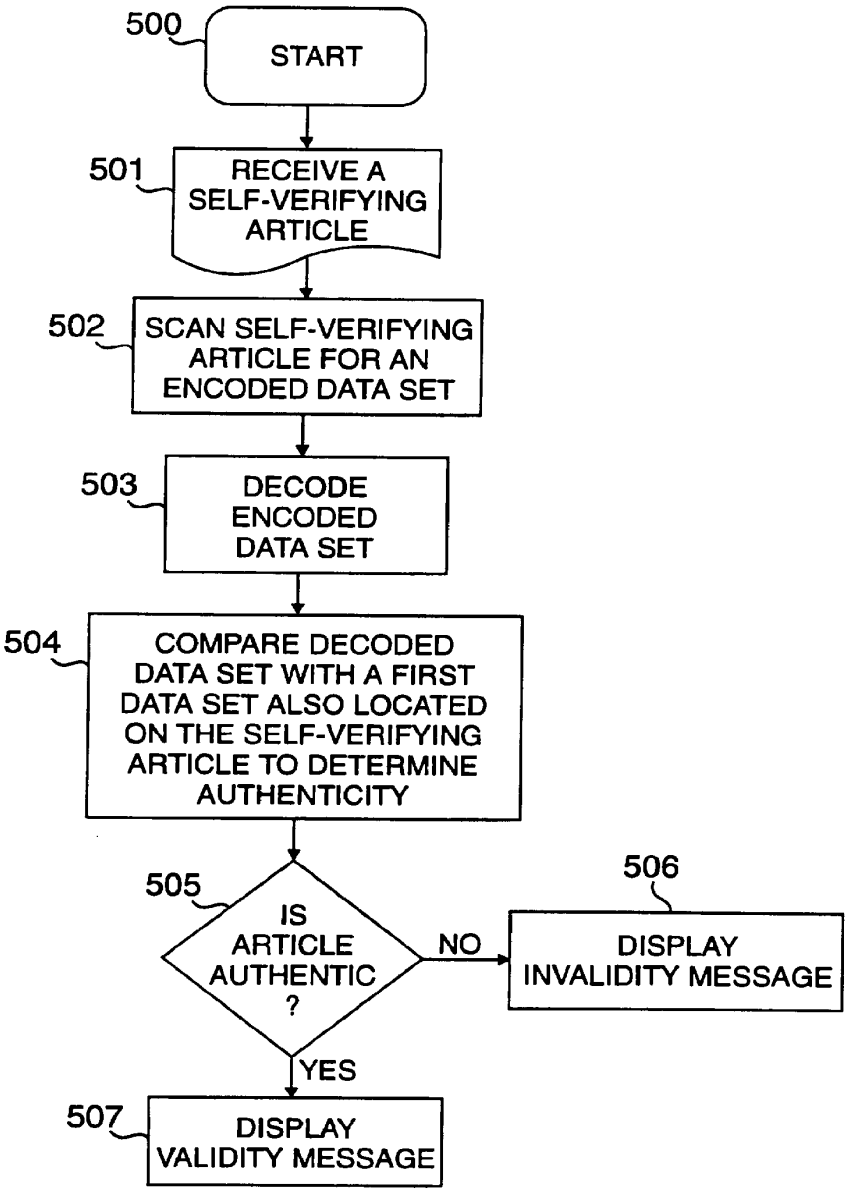

500
START
501
RECEIVE A SELF-VERIFYING ARTICLE
502
SCAN SELF-VERIFYING ARTICLE FOR AN ENCODED DATA SET
503
DECODE ENCODED DATA SET
504
COMPARE DECODED DATA SET WITH A FIRST DATA SET ALSO LOCATED ON THE SELF-VERIFYING ARTICLE TO DETERMINE AUTHENTICITY
505
IS ARTICLE AUTHENTIC ?
NO
506
DISPLAY INVALIDITY MESSAGE
YES
507
DISPLAY VALIDITY MESSAGE

FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US95/09398

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :B42D 15/10
US CL :283/72, 75, 82, 93, 107, 58, 59, 901, 904
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 283/72, 74, 75, 77, 82, 93, 107-112, 57-59, 901, 904; 235/380, 487, 488

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X, P ------- Y, P	WO, A, 94/19770 (HER MAJESTY IN RIGHT OF CANADA) 01 September 1994, see entire document.	1-4, 6, 9, 11, 13-15, 17 ---------- 5, 7, 8, 10, 12, 16, 18-21
A	EP, A, 560,318 (OMIRON CORPORATION) 15 September 1993, see entire document.	

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 30 SEPTEMBER 1995	Date of mailing of the international search report 17 OCT 1995
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer FRANCES HAN Telephone No. (703) 308-1148

Sheila Veney
Paralegal Specialist
Group 3200

Form PCT/ISA/210 (second sheet)(July 1992)*

Electronic Patent Application Fee Transmittal

Application Number:	15824087
Filing Date:	28-Nov-2017
Title of Invention:	APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/ VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG (S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL
First Named Inventor/Applicant Name:	Michael F. Malone
Filer:	Keith D. Harden/Paula Sandu
Attorney Docket Number:	MPOR60-33823

Filed as Large Entity

Filing Fees for Utility under 35 USC 111(a)

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
SUBMISSION- INFORMATION DISCLOSURE STMT	1806	1	240	240
		Total in USD ($)		**240**

Electronic Acknowledgement Receipt

EFS ID:	32418922
Application Number:	15824087
International Application Number:	
Confirmation Number:	5197
Title of Invention:	APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/ VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG (S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL
First Named Inventor/Applicant Name:	Michael F. Malone
Customer Number:	25883
Filer:	Keith D. Harden
Filer Authorized By:	
Attorney Docket Number:	MPOR60-33823
Receipt Date:	23-APR-2018
Filing Date:	28-NOV-2017
Time Stamp:	18:42:51
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	DA
Payment was successfully received in RAM	$240
RAM confirmation Number	042418INTEFSW00009567200780
Deposit Account	200780
Authorized User	Keith Harden

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

37 CFR 1.17 (Patent application and reexamination processing fees)

37 CFR 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Information Disclosure Statement (IDS) Form (SB08)	MPO6033823IDS1.pdf	1043927 98c2d84fd8fd7c7053da8030b98cce330de01780	no	34
Warnings:					
Information:					
2	Information Disclosure Statement (IDS) Form (SB08)	MPO6033823IDS2.pdf	612400 e06aa12a4e7556ed89249e7d31681e9eb2f4d4e7	no	4
Warnings:					
Information:					
3	Foreign Reference	EP0905966Canon.pdf	2643381 7f53d87a676543f549d3d7202bffcd1789c1af4d	no	51
Warnings:					
Information:					
4	Foreign Reference	WO0152178Digimarc.pdf	2344807 82db5a335b310d1a2732ac894f40d7bc8a2e7bfb	no	48
Warnings:					
Information:					
5	Foreign Reference	WO9901859Imige.pdf	1443070 3079b5cbd090a3ae7ad6a365d92dd4e465798d63	no	35
Warnings:					
Information:					
6	Foreign Reference	WO0167707ScaleEight.pdf	3646808 efdfbbe2afd165e5aefa75f081919c7babbce7b1	no	82
Warnings:					
Information:					

7	Foreign Reference	EP1204277Tektronix.pdf	196334 128c46b72cebad1215f920257ca4850a9afd90cf	no	4
Warnings:					
Information:					
8	Foreign Reference	WO9743736Digimarc.pdf	9976201 25b97c45e09763bde6deb581c66c928c8d49490d	no	208
Warnings:					
Information:					
9	Foreign Reference	WO9626494TheRegents.pdf	1614345 dc216794a524589bf76d95bc75b00f1a93bf205b	no	54
Warnings:					
Information:					
10	Foreign Reference	WO9936876Digimarc.pdf	652081 2b66c6a34a2648d138317fbcb48490d412ff20f2	no	22
Warnings:					
Information:					
11	Foreign Reference	WO0115021DigitalConvergence.pdf	2813809 5c0944a242240e56b88cdbd1dee0e126febe91af	no	63
Warnings:					
Information:					
12	Foreign Reference	WO0161508Digimarc.pdf	4184832 04d53c64b96723ffa2ca0315be21e46bd663be86	no	68
Warnings:					
Information:					
13	Foreign Reference	WO0173586DigitalConvergence.pdf	3161079 3be05cde233d284407fd6d99a112188b7b78f51a	no	66
Warnings:					
Information:					

14	Foreign Reference	WO0175629Neomedia.pdf	1376417 2aebd95f8bf0c718caeb3a0f79ee580614deb444	no	41
Warnings:					
Information:					
15	Foreign Reference	WO0197128Markany.pdf	1591510 9a66f2b3f3b61b16ad83888357d1a0542579e0f5	no	42
Warnings:					
Information:					
16	Foreign Reference	WO0211446Koninklijke.pdf	1508421 610b2936bbdcf79befdbaf0146fdbcf25fbbc3f4	no	33
Warnings:					
Information:					
17	Foreign Reference	WO9827510TheRegents.pdf	2073262 c88ce9d748acb008d5bc67d308dce62c88d17ddd	no	51
Warnings:					
Information:					
18	Foreign Reference	WO0036605KentRidge.pdf	1160564 7de777eeadfb77e6bb0a2a13e0846fe02f75e2d4	no	27
Warnings:					
Information:					
19	Foreign Reference	WO0070585Digimarc.pdf	6706219 9644df385a97b7dd6c7998881bf506fdcc0fb9ac	no	137
Warnings:					
Information:					
20	Foreign Reference	WO9603286InternationalData Matrix.pdf	1512473 e47458efe9652eadcc3339979d13796bc77bfb4e	no	34
Warnings:					
Information:					

21	Non Patent Literature	USPTO_187pp_16Feb2015.pdf	11107153 0a5de209b3e5e78942f549f0176e930765e64f56	no	187
Warnings:					
Information:					
22	Non Patent Literature	USPTO_177pp_2March2015.pdf	10288734 022ebd5145b7441155348ae3ef9659fc7c730d6a	no	177
Warnings:					
Information:					
23	Non Patent Literature	KeyedHashing_Feb1997.pdf	400094 52744adf30fc03bb67c35f403769b8e2e0b7c339	no	10
Warnings:					
Information:					
24	Non Patent Literature	TheTrustworthyDigitalCamera_Nov1993.pdf	415107 a6194dbcb4407b26b897db2a2999d33e8e71e662	no	6
Warnings:					
Information:					
25	Non Patent Literature	HTTPAuthenticationBasicandDigestAccess_June1999.pdf	1323840 6acd1b324422c5d39d890faa325206aeb8f6784d	no	34
Warnings:					
Information:					
26	Non Patent Literature	AnExtensiontoHTTP_Jan1997.pdf	688536 3c0ad8b78ace3d755cd21a9759f621ec34df94c5	no	18
Warnings:					
Information:					
27	Non Patent Literature	SuppFirstAmendedComplaint_8Mar2012.pdf	447769 6cd651219264ac75b502d0922502be056f778c74	no	11
Warnings:					
Information:					

28	Non Patent Literature	DigitalStillCameraImageFileFormatStandard_173pp.pdf	4370293 a4db94e1258c524d1a31839c87434a1501737830	no	172
Warnings:					
Information:					
29	Non Patent Literature	TIFFSpecificationRevision6_0_3June1992.pdf	4141806 3d062994bc1f6713f519c1946479f1d5e3a9ef8d	no	121
Warnings:					
Information:					
30	Non Patent Literature	FundamentalsOfGPSSystemReceivers_2000.pdf	11420961 2883d2c0c31ca6c74470257e4f54b6023931abae	no	256
Warnings:					
Information:					
31	Non Patent Literature	PPPEncryptionControlProtocol_June1996.pdf	329399 d692fa8a99679885e24f6a9546d2c49c83d2da67	no	11
Warnings:					
Information:					
32	Non Patent Literature	MyPortDisclosurePursuantToPatentRule3_1_13May2011.pdf	1467051 9bda561b7bfbf52ed1df4366a3e3002aa9012348	no	25
Warnings:					
Information:					
33	Non Patent Literature	CivilGPSFromAFuturePerspective_1Oct1983.pdf	631004 57daa994759575ae13ba5ad634cdbf3853211691	no	6
Warnings:					
Information:					
34	Non Patent Literature	TheNeedForAMetaTagStandard_1Jan2002.pdf	90750 1d91ce3bccea5b0966fe4870bbc6a6459a461cec	no	2
Warnings:					
Information:					

35	Non Patent Literature	CollaborativeVideoAnnotation_1Jan2002.pdf	2444114 3a950e8d8d994951c052085f2da8f9aaebce1a09	no	32
Warnings:					
Information:					
36	Non Patent Literature	USPTO_906pp_27May2014_resized.pdf	9357032 90240d7cdd4449afb769dd38c3eb598624dc22eb	no	906
Warnings:					
Information:					
37	Non Patent Literature	USPTO_1085pp_6June2014_resized.pdf	10027364 7f24c85eab73d8d5504f8a46b4ead83854f844d9	no	1085
Warnings:					
Information:					
38	Non Patent Literature	HandbookOfAppliedCriptography_pp1_397.pdf	12665583 31e3b9d537e0048c0c207fa586a569d072f0f8d8	no	397
Warnings:					
Information:					
39	Non Patent Literature	HandbookOfAppliedCriptography_pp398_794.pdf	14353851 9252e921d77df472a571f3000d73a540d8a1be77	no	397
Warnings:					
Information:					
40	Fee Worksheet (SB06)	fee-info.pdf	30825 528c4edb47c29f69028b1ae9cc70168c6fb3e6c4	no	2
Warnings:					
Information:					
			Total Files Size (in bytes):	146263206	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
15/824,087	11/28/2017	Michael F. Malone	MPOR60-33823

CONFIRMATION NO. 5197

25883
MUNCK WILSON MANDALA L.L.P
P.O. Drawer 800889
DALLAS, TX 75380

PUBLICATION NOTICE

OC000000098228847

Title:APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL

Publication No.US-2018-0083779-A1
Publication Date:03/22/2018

NOTICE OF PUBLICATION OF APPLICATION

The above-identified application will be electronically published as a patent application publication pursuant to 37 CFR 1.211, et seq. The patent application publication number and publication date are set forth above.

The publication may be accessed through the USPTO's publically available Searchable Databases via the Internet at www.uspto.gov. The direct link to access the publication is currently http://www.uspto.gov/patft/.

The publication process established by the Office does not provide for mailing a copy of the publication to applicant. A copy of the publication may be obtained from the Office upon payment of the appropriate fee set forth in 37 CFR 1.19(a)(1). Orders for copies of patent application publications are handled by the USPTO's Public Records Division. The Public Records Division can be reached by telephone at (571) 272-3150 or (800) 972-6382, by facsimile at (571) 273-3250, by mail addressed to the United States Patent and Trademark Office, Public Records Division, Alexandria, VA 22313-1450 or via the Internet.

In addition, information on the status of the application, including the mailing date of Office actions and the dates of receipt of correspondence filed in the Office, may also be accessed via the Internet through the Patent Electronic Business Center at www.uspto.gov using the public side of the Patent Application Information and Retrieval (PAIR) system. The direct link to access this status information is currently https://portal.uspto.gov/pair/PublicPair. Prior to publication, such status information is confidential and may only be obtained by applicant using the private side of PAIR.

Further assistance in electronically accessing the publication, or about PAIR, is available by calling the Patent Electronic Business Center at 1-866-217-9197.

Office of Data Managment, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
15/824,087	11/28/2017	Michael F. Malone	MPOR60-33823	5197

25883 7590 03/08/2018
MUNCK WILSON MANDALA L.L.P
P.O. Drawer 800889
DALLAS, TX 75380

EXAMINER	
SHERALI, ISHRAT I	

ART UNIT	PAPER NUMBER
2667	

NOTIFICATION DATE	DELIVERY MODE
03/08/2018	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

admin@dalpat.com
eofficeaction@appcoll.com
IPdocketing@munckwilson.com

Office Action Summary	Application No. 15/824,087	Applicant(s) MALONE, MICHAEL F.	
	Examiner ISHRAT I. SHERALI	Art Unit 2667	AIA (First Inventor to File) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

5)☒ Claim(s) *1-20* is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6)☐ Claim(s) _____ is/are allowed.

7)☒ Claim(s) *1-20* is/are rejected.

8)☐ Claim(s) _____ is/are objected to.

9)☐ Claim(s) _____ are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a)☐ All b)☐ Some** c)☐ None of the:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b) Paper No(s)/Mail Date _____.

3) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

4) ☐ Other: _____.

Notice of Pre-AIA or AIA Status

1. The present application is being examined under the pre-AIA first to invent provisions.

Claim Objection

2. Claims 1-20 are objected because independent claims 1 and 20 are directed toward data which is not any of the four eligible categories of invention. Claims 1 and 20 do not recite in the preamble and body of the claims image /audio data or any other kind of data. Based on the broadest reasonable interpretation, claims are directed processing of data instead of image/audio/voice data . Claims 1 and 20 are not directed toward machine, manufacture or composition of matter because those categories require a tangible object which is not satisfied by pure data.

3. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the “right to exclude” granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed.

Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on nonstatutory double patenting provided the reference application or patent either is shown to be commonly owned with the examined application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement. See MPEP § 717.02 for applications subject to examination under the first inventor to file provisions of the AIA as explained in MPEP § 2159. See MPEP §§ 706.02(l)(1) - 706.02(l)(3) for applications not subject to examination under the first inventor to file provisions of the AIA. A terminal disclaimer must be signed in compliance with 37 CFR 1.321(b).

The USPTO Internet website contains terminal disclaimer forms which may be used. Please visit www.uspto.gov/patent/patents-forms. The filing date of the application in which the form is filed determines what form (e.g., PTO/SB/25, PTO/SB/26, PTO/AIA/25, or PTO/AIA/26) should be used. A web-based eTerminal Disclaimer may be filled out completely online using web-screens. An eTerminal Disclaimer that meets all requirements is auto-processed and approved immediately upon submission. For more information about eTerminal Disclaimers, refer to www.uspto.gov/patents/process/file/efs/guidance/eTD-info-I.jsp.

4. Claims 1-20 are rejected on the ground of non-statutory obviousness-type double patenting as being un-patentable over claims 1-12 of U.S. Patent No. 7,778,438.

Although the conflicting claims are not identical, they are not patentably distinct from each other because they claiming the same subject matter of converting information in another format and storing in the database and also independent claim 1 of the U.S Patent No.7,778,438 includes all of the limitations recited in independent claims 1 and 20 of the instant application, except claims 1 and 20 of the instant application also include "transmitting defined set of first captured information and the defined set of stored captured information to a remote location . Since independent claim 1 of the U.S Patent No 7,778,438 recites in the preamble "comprising" i.e. claim 1 of the U.S Patent No 7,778,438 is open- ended claim therefore it also include transmitting defined set of first captured information and the defined set of stored captured information to a remote location and furthermore independent claim 1 of the U.S Patent No 7,778,438 also recite captured information and remote database.

Similarly claims 2-19 of the instant application are similarly obvious over claims 1-12 of the U.S. Patent No. 7,778,438.

Communication

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sherali Ishrat whose telephone number is 571-272-7398. The examiner can normally be reached on 8:00 AM - 4:30PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matthew Bella can be reached on 571-272-7453. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/ISHRAT I SHERALI/
Primary Examiner, Art Unit 2667

March 1, 2018

Notice of References Cited	Application/Control No. 15/824,087	Applicant(s)/Patent Under Reexamination MALONE, MICHAEL F.	
	Examiner ISHRAT I. SHERALI	Art Unit 2667	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	CPC Classification	US Classification
*	A	US-8,983,119 B2	03-2015	Malone; Michael F.	G06F17/30038	382/100
*	B	US-8,768,693 B2	07-2014	Somekh; Oren	G06F17/30265	348/231.4
*	C	US-8,509,477 B2	08-2013	Malone; Michael F.	G06F17/30038	382/100
*	D	US-7,778,438 B2	08-2010	Malone; Michael F.	G06F17/30038	382/100
*	E	US-7,043,048 B1	05-2006	Ellingson; Eric E.	H04N1/32101	382/100
*	F	US-6,990,444 B2	01-2006	Hind; John R.	G10L21/06	704/235
*	G	US-6,687,383 B1	02-2004	Kanevsky; Dimitri	G10L19/018	380/210
*	H	US-9,471,592 B2	10-2016	Koyama; Toshiaki	H04L12/1831	1/1
*	I	US-9,832,017 B2	11-2017	Malone; Michael F.	H04L9/14	1/1
	J	US-				
	K	US-				
	L	US-				
	M	US-				

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	CPC Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Bibliographic Data

Application No: 15824087

Foreign Priority claimed: ○ Yes ◉ No

35 USC 119 (a-d) conditions met: ☐ Yes ☑ No ☐ Met After Allowance

Verified and Acknowledged: /ISHRAT SHERALI/ Examiner's Signature | Initials

Title: APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL

FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
11/28/2017	382	2667	MPOR60-33823
RULE			

APPLICANTS

MYPORT TECHNOLOGIES, INC., McKinney, TX,

INVENTORS

Michael F. Malone, McKinney, TX, UNITED STATES

CONTINUING DATA

This application is a CON of 15272013 09/21/2016 PAT 9832017

15272013 is a CON of 14660166 03/17/2015

14660166 is a CON of 13965625 08/13/2013 PAT 8983119

13965625 is a CON of 13417229 03/10/2012 PAT 8509477

13417229 is a CON of 12857358 08/16/2010 PAT 8135169

12857358 is a CON of 11621062 01/08/2007 PAT 7778438

11621062 has PRO of 60757075 01/06/2006

11621062 is a CIP of 11325373 01/04/2006 PAT 7184573

11325373 is a CON of 10674910 09/29/2003 PAT 6996251

10674910 has PRO of 60414449 09/30/2002

FOREIGN APPLICATIONS

IF REQUIRED, FOREIGN LICENSE GRANTED**

12/08/2017

STATE OR COUNTRY

UNITED STATES

ADDRESS

MUNCK WILSON MANDALA L.L.P

P.O. Drawer 800889
DALLAS, TX 75380
UNITED STATES

FILING FEE RECEIVED

$1,600

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	22471	(captur$3 or record$3) with (audio or speech) same conver$8	USPAT	OR	OFF	2018/03/04 22:54
L2	116642	(captur$3 or record$3 or camera) with (voice or audio or speech or image or text) same (conver$8 or convert$3)	USPAT	OR	ON	2018/03/04 22:56
L3	11540	(captur$3 or record$3 or camera) with (voice or audio or speech or image or text) same (conver$8 or convert$3) with format	USPAT	OR	ON	2018/03/04 22:57
L4	119142	(captur$3 or record$3 or camera) with (voice or audio or speech or image or text) with time	USPAT	OR	ON	2018/03/04 22:59
L5	435	(captur$3 or record$3 or camera) with (voice or audio or speech or image or text) with time same transmit$6 same (stor$5 or archiv$3) with (database or librar$3)	USPAT	OR	ON	2018/03/04 23:01
L6	728795	(captur$3 or record$3 or camera) with (voice or audio or speech or image or text) with time same transmit$6 same (stor$5 or archiv$3) with (database or librar$3) same conver$6 with media or format	USPAT	OR	ON	2018/03/04 23:02
L7	1	(captur$3 or record$3 or camera) with (voice or audio or speech or image or text) with time same transmit$6 same (stor$5 or archiv$3) with (database or librar$3) same conver$6 with media with format	USPAT	OR	ON	2018/03/04 23:02
L8	6	(captur$3 or record$3 or camera) with (voice or audio or speech or image or text) with time same transmit$6 same (stor$5 or archiv$3) with (database or librar$3) same conver$6 with text	USPAT	OR	ON	2018/03/04 23:03
L9	9	(captur$3 or record$3 or camera) with (voice or audio or speech or image or text) with time same transmit$6 same (stor$5 or archiv$3) with (database or librar$3) same conver$6 with (text or media or format)	USPAT	OR	ON	2018/03/04 23:05
L10	28	(captur$3 or record$3 or camera) with (voice or audio or speech or image or text) with time same transmit$6 same (stor$5 or archiv$3) with (database or librar$3) same conver$6 with (text or media or format)	US-PGPUB; USPAT	OR	ON	2018/03/04 23:05
L11	37	(captur$3 or record$3 or camera) with (voice or audio or speech or image or text) with time same transmit$6 same	US-PGPUB; USPAT; USOCR;	OR	ON	2018/03/04 23:08

		(stor$5 or archiv$3) with (database or librar$3) same (conversion or convert$3) with (text or media or format)	FPRS; EPO; JPO; DERWENT; IBM_TDB			
L12	8	11 AND ((G06F17/30749 OR G06F17/30778 OR G06F17/30887 OR G06F3/165 OR G06F17/30038 OR G06F17/30244 OR G06F17/30268 OR G06F17/3028 OR G06F17/30011 OR G06F17/30253 OR G06F17/30781 OR G06F3/147 OR G06F9/542 OR H04N21/432 OR H04N21/25816 OR H04N21/25875 OR H04N21/26603 OR H04N21/26613 OR H04N21/2747 OR H04N21/6582 OR H04N21/835 OR H04N21/84).CPC.)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2018/03/04 23:11

EAST Search History (Interference)

<This search history is empty>

3/4/2018 11:12:34 PM

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	15824087	MALONE, MICHAEL F.
	Examiner	**Art Unit**
	ISHRAT I SHERALI	2667

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☒ Claims renumbered in the same order as presented by applicant ☐ CPA ☒ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	03/01/2018								
	1	✓								
	2	✓								
	3	✓								
	4	✓								
	5	✓								
	6	✓								
	7	✓								
	8	✓								
	9	✓								
	10	✓								
	11	✓								
	12	✓								
	13	✓								
	14	✓								
	15	✓								
	16	✓								
	17	✓								
	18	✓								
	19	✓								
	20	✓								

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	15824087	MALONE, MICHAEL F.
	Examiner	Art Unit
	ISHRAT I SHERALI	2667

CPC- SEARCHED		
Symbol	Date	Examiner

CPC COMBINATION SETS - SEARCHED		
Symbol	Date	Examiner

US CLASSIFICATION SEARCHED			
Class	Subclass	Date	Examiner
382	100, 305	3/1/18	IS

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

SEARCH NOTES		
Search Notes	Date	Examiner
SEARCH EAST TEXT LIMITED CPC/USPC, INVENTOR/ASSINGEE, GOOGLE	3/1/18	IS
SEARCH EAST TEXT LIMITED FOLLOWING CPC CLASS/SUBCLASS: G06F 17/30038, 30749, 30268, 30244, H04L 9/14, H04N 5/76, 44, H04N 21/25816, 25875, 26603, 26613	3/1/18	IS

INTERFERENCE SEARCH			
US Class/ CPC Symbol	US Subclass / CPC Group	Date	Examiner
	SEARCH INTERFERENCE TEXT LIMITED CPC/USPC	3/1/18	IS

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number: 15/824,087

APPLICATION AS FILED - PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY FEE($)
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	280
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	600
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	720
TOTAL CLAIMS (37 CFR 1.16(i))	20 minus 20 =	*			OR	x 80 =	0.00
INDEPENDENT CLAIMS (37 CFR 1.16(h))	2 minus 3 =	*				x 420 =	0.00
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						0.00
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							0.00
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	1600

APPLICATION AS AMENDED - PART II

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY ADDITIONAL FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE($)
AMENDMENT A	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE($)	ADDITIONAL FEE($)	OR	RATE($)	ADDITIONAL FEE($)
AMENDMENT B	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest found in the appropriate box in column 1.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
15/824,087	11/28/2017	2852	1600	MPOR60-33823	20	2

CONFIRMATION NO. 5197

25883
HOWISON & ARNOTT, L.L.P
P.O. BOX 741715
DALLAS, TX 75374-1715

FILING RECEIPT

OC000000093992239

Date Mailed: 12/12/2017

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Inventor(s)

Michael F. Malone, McKinney, TX;

Applicant(s)

MYPORT TECHNOLOGIES, INC., McKinney, TX;

Power of Attorney: The patent practitioners associated with Customer Number 25883

Domestic Priority data as claimed by applicant

This application is a CON of 15/272,013 09/21/2016 PAT 9832017
which is a CON of 14/660,166 03/17/2015 ABN
which is a CON of 13/965,625 08/13/2013 PAT 8983119
which is a CON of 13/417,229 03/10/2012 PAT 8509477
which is a CON of 12/857,358 08/16/2010 PAT 8135169
which is a CON of 11/621,062 01/08/2007 PAT 7778438
which is a CIP of 11/325,373 01/04/2006 PAT 7184573
and claims benefit of 60/757,075 01/06/2006
and said 11/325,373 01/04/2006
is a CON of 10/674,910 09/29/2003 PAT 6996251
which claims benefit of 60/414,449 09/30/2002

Foreign Applications for which priority is claimed (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see http://www.uspto.gov for more information.) - None.
Foreign application information must be provided in an Application Data Sheet in order to constitute a claim to foreign priority. See 37 CFR 1.55 and 1.76.

Permission to Access Application via Priority Document Exchange: Yes

Permission to Access Search Results: Yes

Applicant may provide or rescind an authorization for access using Form PTO/SB/39 or Form PTO/SB/69 as appropriate.

If Required, Foreign Filing License Granted: 12/08/2017

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 15/824,087**

Projected Publication Date: 03/22/2018

Non-Publication Request: No

Early Publication Request: No

Title

APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL

Preliminary Class

396

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications: No

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4258).

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor

community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop technology, manufacture products, deliver services, and grow your business, visit http://www.SelectUSA.gov or call +1-202-482-6800.

Doc Code: PA..
Document Description: Power of Attorney

PTO/AIA/82A (07-13)
Approved for use through 11/30/2014. OMB 0651-0051
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FOR POWER OF ATTORNEY TO ONE OR MORE REGISTERED PRACTITIONERS

NOTE: This form is to be submitted with the Power of Attorney by Applicant form (PTO/AIA/82B) to identify the application to which the Power of Attorney is directed, in accordance with 37 CFR 1.5, unless the application number and filing date are identified in the Power of Attorney by Applicant form. If neither form PTO/AIA/82A nor form PTO/AIA82B identifies the application to which the Power of Attorney is directed, the Power of Attorney will not be recognized in the application.

Application Number	
Filing Date	
First Named Inventor	Michael F. MALONE
Title	APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL
Art Unit	
Examiner Name	
Attorney Docket Number	MPOR60-33823

SIGNATURE of Applicant or Patent Practitioner

Signature	/Keith D. Harden, Reg. #74472/	Date (Optional)	
Name	Keith D. Harden	Registration Number	74,472
Title (if Applicant is a juristic entity)			
Applicant Name (if Applicant is a juristic entity)			

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4(d) for signature requirements and certifications. If more than one applicant, use multiple forms.

☑ *Total of ONE forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Doc Code: PA..
Document Description: Power of Attorney

PTO/AIA/82B (07-13)
Approved for use through 11/30/2014. OMB 0651-0051
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

POWER OF ATTORNEY BY APPLICANT

I hereby revoke all previous powers of attorney given in the application identified in either the attached transmittal letter or the boxes below.

Application Number	Filing Date

(Note: The boxes above may be left blank if information is provided on form PTO/AIA/82A.)

☑ I hereby appoint the Patent Practitioner(s) associated with the following Customer Number as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above: 25883

OR

☐ I hereby appoint Practitioner(s) named in the attached list (form PTO/AIA/82C) as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the patent application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above. (Note: Complete form PTO/AIA/82C.)

Please recognize or change the correspondence address for the application identified in the attached transmittal letter or the boxes above to:

☑ The address associated with the above-mentioned Customer Number

OR

☐ The address associated with Customer Number:

OR

Firm or Individual Name					
Address					
City		State		Zip	
Country					
Telephone		Email			

I am the Applicant (if the Applicant is a juristic entity, list the Applicant name in the box):

MYPORT TECHNOLOGIES, INC.

☐ Inventor or Joint Inventor (title not required below)

☐ Legal Representative of a Deceased or Legally Incapacitated Inventor (title not required below)

☑ Assignee or Person to Whom the Inventor is Under an Obligation to Assign (provide signer's title if applicant is a juristic entity)

☐ Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.46(b)(2) was granted in the application or is concurrently being filed with this document) (provide signer's title if applicant is a juristic entity)

SIGNATURE of Applicant for Patent

The undersigned (whose title is supplied below) is authorized to act on behalf of the applicant (e.g., where the applicant is a juristic entity).

Signature		Date (Optional)	
Name	Michael F. MALONE		
Title	CEO		

NOTE: Signature - This form must be signed by the applicant in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. If more than one applicant, use multiple forms.

☑ Total of ONE forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/AIA/01 (06-12)
Approved for use through 01/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

DECLARATION (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN APPLICATION DATA SHEET (37 CFR 1.76)

Title of Invention	METHOD FOR VOICE COMMAND ACTIVATION, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH CONVERSION, METATAGS CREATION, STORAGE AND SEARCH RETRIEVAL

As the below named inventor, I hereby declare that:

This declaration is directed to:

- [x] The attached application, or
- [ ] United States application or PCT international application number _____

filed on _____.

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

LEGAL NAME OF INVENTOR

Inventor: Michael E. MALONE Date (Optional): 8/12/13

Signature: _____

Note: An application data sheet (PTO/AIA/14 or equivalent), including naming the entire inventive entity, must accompany this form. Use an additional PTO/SB/AIA01 form for each additional inventor.

This collection of information is required by 35 U.S.C. 115 and 37 CFR 1.63. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 1 minute to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Electronic Patent Application Fee Transmittal

Application Number:	
Filing Date:	
Title of Invention:	APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/ VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG (S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL
First Named Inventor/Applicant Name:	Michael F. Malone
Filer:	Keith D. Harden/Paula Sandu
Attorney Docket Number:	MPOR60-33823

Filed as Large Entity

Filing Fees for Utility under 35 USC 111(a)

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
UTILITY APPLICATION FILING	1011	1	280	280
UTILITY SEARCH FEE	1111	1	600	600
UTILITY EXAMINATION FEE	1311	1	720	720
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
		Total in USD ($)		**1600**

Electronic Acknowledgement Receipt

EFS ID:	31058851
Application Number:	15824087
International Application Number:	
Confirmation Number:	5197
Title of Invention:	APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/ VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG (S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL
First Named Inventor/Applicant Name:	Michael F. Malone
Customer Number:	25883
Filer:	Keith D. Harden
Filer Authorized By:	
Attorney Docket Number:	MPOR60-33823
Receipt Date:	28-NOV-2017
Filing Date:	
Time Stamp:	14:17:53
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	DA
Payment was successfully received in RAM	$1600
RAM confirmation Number	112917INTEFSW00000663200780
Deposit Account	200780
Authorized User	Keith Harden

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

37 CFR 1.16 (National application filing, search, and examination fees)

37 CFR 1.17 (Patent application and reexamination processing fees)

37 CFR 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Application Data Sheet	MPO6033823ADS.pdf	1824060 d0b5a6bf26341b60029809c31cbf80c8e332126b	no	9
Warnings:					
Information:					
2		MPO6033823CONTApp.pdf	157683 c2e1c0a5da27db82b7ed15fdea1847539ed2a178	yes	25

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Abstract	25	25
	Claims	20	24
	Specification	1	19

Warnings:					
Information:					
3	Drawings-only black and white line drawings	MPO6033823FRMLDWGS.pdf	230372 812db9ad4fb625f50c42634785f8f03da5e61bff	no	4
Warnings:					
Information:					
4	Power of Attorney	MPO6033823POAEXE.pdf	1479160 179a9aeb308aaec0d04a4a3f134ce7ccad1e9c3a	no	2
Warnings:					
Information:					
5	Oath or Declaration filed	MPO6033823DECEXE.pdf	149364 4da668fd59aec34c1a34a2e5180af1214b53f861	no	1

Warnings:					
Information:					
6	Fee Worksheet (SB06)	fee-info.pdf	35185 6169c3f7a562767b53cfbe105e9f028f2ed40da6	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	3875824		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	MPOR60-33823
	Application Number	
Title of Invention	APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH	

The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76.
This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.

Secrecy Order 37 CFR 5.2:

☐ Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)

Inventor Information:

Inventor 1 Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Michael	F.	Malone	

Residence Information (Select One) ● US Residency Non US Residency Active US Military Service

City	McKinney	State/Province	TX	Country of Residence	US

Mailing Address of Inventor:

Address 1	2711 Northview Drive		
Address 2			
City	McKinney	State/Province	TX
Postal Code	75070	Country	US

All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the **Add** button. Add

Correspondence Information:

Enter either Customer Number or complete the Correspondence Information section below.
For further information see 37 CFR 1.33(a).

☐ **An Address is being provided for the correspondence Information of this application.**

Customer Number	25883		
Email Address	admin@dalpat.com	Add Email	Remove Email

Application Information:

Title of the Invention	APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL		
Attorney Docket Number	MPOR60-33823	Small Entity Status Claimed	☐
Application Type	Nonprovisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)		Suggested Figure for Publication (if any)	

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	MPOR60-33823
	Application Number	
Title of Invention	APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH	

Filing By Reference:

Only complete this section when filing an application by reference under 35 U.S.C. 111(c) and 37 CFR 1.57(a). Do not complete this section if application papers including a specification and any drawings are being filed. Any domestic benefit or foreign priority information must be provided in the appropriate section(s) below (i.e., "Domestic Benefit/National Stage Information" and "Foreign Priority Information").

For the purposes of a filing date under 37 CFR 1.53(b), the description and any drawings of the present application are replaced by this reference to the previously filed application, subject to conditions and requirements of 37 CFR 1.57(a).

Application number of the previously filed application	Filing date (YYYY-MM-DD)	Intellectual Property Authority or Country

Publication Information:

☐ Request Early Publication (Fee required at time of Request 37 CFR 1.219)

☐ **Request Not to Publish.** I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application **has not and will not** be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

Representative Information:

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32).
Either enter Customer Number or complete the Representative Name section below. If both sections are completed the customer Number will be used for the Representative Information during processing.

Please Select One:	● Customer Number	US Patent Practitioner	◯ Limited Recognition (37 CFR 11.9)
Customer Number	25883		

Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, 365(c), or 386(c) or indicate National Stage entry from a PCT application. Providing benefit claim information in the Application Data Sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78.
When referring to the current application, please leave the "Application Number" field blank.

Prior Application Status: Patented — Remove

Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
	Continuation of	15/272013	2016-09-21	9832017	2017-11-28

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	MPOR60-33823
	Application Number	
Title of Invention	APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH	

Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
15/272013	Continuation of	14/660166	2015-03-17

Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
14/660166	Continuation of	13/965625	2013-08-13	8983119	2015-03-17

Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
13/965625	Continuation of	13/417229	2012-03-10	8509477	2013-08-13

Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
13/417229	Continuation of	12/857358	2010-08-16	8135169	2012-03-13

Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
12/857358	Continuation of	11/621062	2007-01-08	7778438	2010-08-17

Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
11/621062	Continuation in part of	11/325373	2006-01-04	7184573	2007-02-27

Prior Application Status	Expired		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
11/621062	Claims benefit of provisional	60/757075	2006-01-06

Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
11/325373	Continuation of	10/674910	2003-09-29	6996251	2006-02-07

Prior Application Status	Expired		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
10/674910	Claims benefit of provisional	60/414449	2002-09-30

Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the **Add** button. Add

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	MPOR60-33823
	Application Number	
Title of Invention	APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH	

Foreign Priority Information:

This section allows for the applicant to claim priority to a foreign application. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55. When priority is claimed to a foreign application that is eligible for retrieval under the priority document exchange program (PDX)[i] the information will be used by the Office to automatically attempt retrieval pursuant to 37 CFR 1.55(i)(1) and (2). Under the PDX program, applicant bears the ultimate responsibility for ensuring that a copy of the foreign application is received by the Office from the participating foreign intellectual property office, or a certified copy of the foreign priority application is filed, within the time period specified in 37 CFR 1.55(g)(1).

Remove

Application Number	Country[i]	Filing Date (YYYY-MM-DD)	Access Code[i] (if applicable)

Additional Foreign Priority Data may be generated within this form by selecting the **Add** button. Add

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications

☐ This application (1) claims priority to or the benefit of an application filed before March 16, 2013 and (2) also contains, or contained at any time, a claim to a claimed invention that has an effective filing date on or after March 16, 2013.
NOTE: By providing this statement under 37 CFR 1.55 or 1.78, this application, with a filing date on or after March 16, 2013, will be examined under the first inventor to file provisions of the AIA.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	MPOR60-33823
	Application Number	
Title of Invention	APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH	

Authorization or Opt-Out of Authorization to Permit Access:

When this Application Data Sheet is properly signed and filed with the application, applicant has provided written authority to permit a participating foreign intellectual property (IP) office access to the instant application-as-filed (see paragraph A in subsection 1 below) and the European Patent Office (EPO) access to any search results from the instant application (see paragraph B in subsection 1 below).

Should applicant choose not to provide an authorization identified in subsection 1 below, applicant **must opt-out** of the authorization by checking the corresponding box A or B or both in subsection 2 below.

NOTE: This section of the Application Data Sheet is **ONLY** reviewed and processed with the **INITIAL** filing of an application. After the initial filing of an application, an Application Data Sheet cannot be used to provide or rescind authorization for access by a foreign IP office(s). Instead, Form PTO/SB/39 or PTO/SB/69 must be used as appropriate.

1. Authorization to Permit Access by a Foreign Intellectual Property Office(s)

A. Priority Document Exchange (PDX) - Unless box A in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the European Patent Office (EPO), the Japan Patent Office (JPO), the Korean Intellectual Property Office (KIPO), the State Intellectual Property Office of the People's Republic of China (SIPO), the World Intellectual Property Organization (WIPO), and any other foreign intellectual property office participating with the USPTO in a bilateral or multilateral priority document exchange agreement in which a foreign application claiming priority to the instant patent application is filed, access to: (1) the instant patent application-as-filed and its related bibliographic data, (2) any foreign or domestic application to which priority or benefit is claimed by the instant application and its related bibliographic data, and (3) the date of filing of this Authorization. See 37 CFR 1.14(h)(1).

B. Search Results from U.S. Application to EPO - Unless box B in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the EPO access to the bibliographic data and search results from the instant patent application when a European patent application claiming priority to the instant patent application is filed. See 37 CFR 1.14(h)(2).

The applicant is reminded that the EPO's Rule 141(1) EPC (European Patent Convention) requires applicants to submit a copy of search results from the instant application without delay in a European patent application that claims priority to the instant application.

2. Opt-Out of Authorizations to Permit Access by a Foreign Intellectual Property Office(s)

☐ A. Applicant **DOES NOT** authorize the USPTO to permit a participating foreign IP office access to the instant application-as-filed. If this box is checked, the USPTO will not be providing a participating foreign IP office with any documents and information identified in subsection 1A above.

☐ B. Applicant **DOES NOT** authorize the USPTO to transmit to the EPO any search results from the instant patent application. If this box is checked, the USPTO will not be providing the EPO with search results from the instant application.

NOTE: Once the application has published or is otherwise publicly available, the USPTO may provide access to the application in accordance with 37 CFR 1.14.

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	MPOR60-33823
	Application Number	
Title of Invention	APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH	

Applicant Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

Applicant 1 — Remove

If the applicant is the inventor (or the remaining joint inventor or inventors under 37 CFR 1.45), this section should not be completed. The information to be provided in this section is the name and address of the legal representative who is the applicant under 37 CFR 1.43; or the name and address of the assignee, person to whom the inventor is under an obligation to assign the invention, or person who otherwise shows sufficient proprietary interest in the matter who is the applicant under 37 CFR 1.46. If the applicant is an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest) together with one or more joint inventors, then the joint inventor or inventors who are also the applicant should be identified in this section.

Clear

● Assignee	Legal Representative under 35 U.S.C. 117	Joint Inventor
Person to whom the inventor is obligated to assign.	Person who shows sufficient proprietary interest	

If applicant is the legal representative, indicate the authority to file the patent application, the inventor is:

Name of the Deceased or Legally Incapacitated Inventor:

If the Applicant is an Organization check here. ☒

Organization Name	MYPORT TECHNOLOGIES, INC.		
Mailing Address Information For Applicant:			
Address 1	2711 Northview Drive		
Address 2			
City	McKinney	**State/Province**	TX
Country	US	Postal Code	75070
Phone Number		Fax Number	
Email Address			

Additional Applicant Data may be generated within this form by selecting the Add button. Add

Assignee Information including Non-Applicant Assignee Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	MPOR60-33823
	Application Number	
Title of Invention	APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH	

Assignee 1

Complete this section if assignee information, including non-applicant assignee information, is desired to be included on the patent application publication. An assignee-applicant identified in the "Applicant Information" section will appear on the patent application publication as an applicant. For an assignee-applicant, complete this section only if identification as an assignee is also desired on the patent application publication.

Remove

If the Assignee or Non-Applicant Assignee is an Organization check here. ☐

Prefix	**Given Name**	Middle Name	**Family Name**	Suffix

Mailing Address Information For Assignee including Non-Applicant Assignee:

Address 1			
Address 2			
City		**State/Province**	
Country i		Postal Code	
Phone Number		Fax Number	
Email Address			

Additional Assignee or Non-Applicant Assignee Data may be generated within this form by selecting the Add button. Add

Signature:

Remove

NOTE: This Application Data Sheet must be signed in accordance with 37 CFR 1.33(b). **However, if this Application Data Sheet is submitted with the INITIAL filing of the application and either box A or B is not checked in subsection 2 of the "Authorization or Opt-Out of Authorization to Permit Access" section, then this form must also be signed in accordance with 37 CFR 1.14(c).**

This Application Data Sheet **must** be signed by a patent practitioner if one or more of the applicants is a **juristic entity** (e.g., corporation or association). If the applicant is two or more joint inventors, this form must be signed by a patent practitioner, **all** joint inventors who are the applicant, or one or more joint inventor-applicants who have been given power of attorney (e.g., see USPTO Form PTO/AIA/81) on behalf of **all** joint inventor-applicants.

See 37 CFR 1.4(d) for the manner of making signatures and certifications.

Signature	/Keith D. Harden, Reg. # 74472/			Date (YYYY-MM-DD)	2017-11-28
First Name	Keith	Last Name	Harden	Registration Number	74,472

Additional Signature may be generated within this form by selecting the Add button. Add

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	MPOR60-33823
	Application Number	
Title of Invention	APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH	

This collection of information is required by 37 CFR 1.76. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1 The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3 A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent CooperationTreaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

ABSTRACT

This invention relates to a network interface device. A first capture device interfaces with a first external information source to capture first external information. A processor processes the captured first external information and stores it in a first media. The processor initiates the storage of the first captured information at an initial time and completes storage of the first captured information at a completion time, thus providing a stored defined set of first captured information. A transmitter transmits the defined set of stored captured information to a remote location on a network. A remote processing system is disposed at the remote node on the network and includes a database and a receiver for receiving the transmitted defined set of first captured information. A data converter is operable to convert the received defined set of first captured information to a second format. The database stores the set of converted captured information.

WHAT IS CLAIMED IS:

1. A system for capturing data in a first media and storing in a database at a location on a network, comprising:

a capture device having:

a first data converter interfacable with a first external information source that generates external information and the first data converter capturing the first external information during generation thereof,

the first data converter processing the captured first external information and storing it in a first media format as stored first captured information within the capture device, the first data converter initiating the storage of the first captured information as stored first captured information at an initial time and completing storage of the first captured information as stored first captured information at a completion time, thus providing a stored defined set of first captured information representing the first captured information between the initial time and the completion time, and

a transmitter for transmitting as a transmitted defined set of first captured information the defined set of stored captured information to the location on the network after the completion time;

a system disposed at the location on the network and including:

a database,

a receiver for receiving the transmitted defined set of first captured information from the transmitter associated with the capture device as a received defined set of first captured information,

a system data converter for converting the received defined set of first captured information to a second format as a set of converted first captured information, the second format different than the first media format, the second format being a searchable format, and

the database storing the set of converted captured information.

2. The system of Claim 1, and further comprising:

a second data converter in the capture device for interfacing with a second external information source to capture second external information therefrom;

the second data converter processing the captured second external information and storing it in a third format different from the first media format as stored second captured information;

a combiner for associating the defined set of stored first captured information with the stored second captured information;

the transmitter transmitting the associated defined set of stored first captured information and stored second captured information to the location on the network;

the receiver at the system receiving the transmitted defined set of stored first captured information and stored second captured information form the transmitter;

the database storing the set of converted captured information with the associated stored second captured information such that the association of the defined set of stored first captured information and stored second captured information is maintained.

3. The system of Claim 2, wherein the second external information is temporal and is complete as to the capture at the time of capture by the second data converter.

4. The system of Claim 2, wherein the third format is a second media format different from the first media format and wherein the third format of a second media format is maintained in the database at the location on the network.

5. The system of Claim 1, wherein the second data converter captures information from the second external information source indicative of the location of the capture device and the second data converter processing the second captured information for storage as location information as the stored second captured information.

6. The system of Claim 5, and further comprising an other data capture device in the capture device that comprises a Global Positioning System (GPS) device.

7. The system of Claim 5, and further comprising an other data capture device in the capture device that derives location information from a time domain analysis of mobile phone signals.

8. The system of Claim 5, and further comprising an other data capture device in the capture device that derives location information from a time domain analysis of wireless network domains.

9. The system of Claim 5, and further comprising an other data capture device in the capture device that derives location information from Doppler analysis of timing signals from cellular/PCS base stations.

10. The system of Claim 5, wherein the capture device comprises a cell based communication device and further comprising an other data capture device in the capture device that captures location information as to cell transmitters and the other data capture device in the capture device that determines location information from cell triangulation.

11. The system of Claim 2, wherein the first data converter comprises an audio transducer and the first media format comprises a format representing audio information.

12. The system of Claim 11, wherein the second format of the converted set of first captured information represents text.

13. The system of Claim 1, wherein the first data converter comprises an audio transducer and the first media format comprises a format representing audio information.

14. The system of Claim 13, wherein the second format of the set of converted first captured information represents text.

15. The system of Claim 1, wherein the capture device comprises a portable handheld device.

16. The system of Claim 1, the capture device has stored thereat a first capture device identifier and the transmitter transmitting the first capture device identifier with the associated defined set of stored first captured information and stored second captured information to the remote location.

17. The system of Claim 1, and further comprising an encryptor for encrypting at least a portion of the defined set of stored captured information transmitted by the transmitter at some time between the completion time and the receipt by the system.

18. The system of Claim 1, wherein the encryptor encrypts the defined set of stored captured information proximate in time to the completion time.

19. The system of Claim 1, wherein the encryptor encrypts at a first level of encryption as an encrypted file that constitutes a symmetrical encryption algorithm that has a secret key available only to an owner of the defined set of stored captured information and wherein the encrypted file is placed in combination with the owner information in the transmission and wherein the system is operable to store the received information.

20. A system for capturing data in a first media for storage in a database at a location on a network, comprising:

a capture device having:

a first data converter interfacable with a first external information source that generates external information and capturing the first external information during generation thereof,

a second data converter interfacing with a second external information source to capture second external information therefrom at an instant in time;

the first data converter processing the captured first external information from a start event representing the time that capture is initiated to a stop event representing the time that capture is complete and storing it in a first media format as a stored defined set of first captured information within the capture device, the second data converter for processing the captured second external information and

storing it as second captured information and a combiner for generating an association between the defined set of first captured information and the second captured information, and

a transmitter for transmitting the associated stored defined set of first captured information and second captured information as a transmitted associated defined set of first captured information and second captured information to the location on the network after at least stop event associated with the associated with the processing of the captured second external information;

a system disposed at the location on the network and including:

a database,

a receiver for receiving the transmitted associated defined set of first captured information and second captured information from the transmitter associated with the capture device as a received set of first captured information,

a system data converter for converting the received defined set of first captured information to a second format as a set of converted first captured information, the second format different than the first media format, the second format being a searchable format, and

the database storing the set of converted captured information in association with the associated received second set of captured information.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
APPLICATION FOR UNITED STATES PATENT

APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL

Inventor:

MALONE, Michael F. – McKinney, Texas

Attorneys:

MUNCK WILSON MANDALA L.L.P.
600 Banner Place
12770 Coit Road
Dallas, Texas 75251
(972) 628-3600

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a Continuation of U.S. Patent Application No. 15/272,013, filed on September 21, 2016, entitled APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL, now U.S. Patent No. 9,832,017, issued November 28, 2017 (Atty. Dkt. No. MPOR-33317). Application No. 15/272,013 is a continuation of U.S. Application No. 14/660,166, filed March 17, 2015, and entitled METHOD FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH CONVERSION, METATAGS CREATION, STORAGE AND SEARCH RETRIEVAL (Atty. Dkt. No. MPOR-32495). U.S. Application No. 14/660,166 is a Continuation of U.S. Patent Application No. 13/965,625, filed August 13, 2013, and entitled METHOD FOR VOICE COMMAND ACTIVATION, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH CONVERSION, METATAGS CREATION, STORAGE AND SEARCH RETRIEVAL, now U.S. Patent No. 8,983,119, issued on March 17, 2015 (Atty. Dkt. No. MPOR-31788). U.S. Patent Application No. 13/965,625 is a Continuation of U.S. Patent Application No. 13/417,229, filed on March 10, 2012, and entitled METHOD FOR MULTI-MEDIA CAPTURE, TRANSMISSION, CONVERSION, METATAGS CREATION, STORAGE AND SEARCH RETRIEVAL, now U.S. Patent No. 8,509,477, issued on August 13, 2013 (Atty. Dkt. No. MPOR-31064), the specifications of which are incorporated herein by reference.

[0002] U.S. Patent Application No. 13/417,229 is a Continuation of U.S. Patent Application No. 12/857,358, filed on August 16, 2010, and entitled METHOD FOR MULTI-MEDIA CONVERSION, METATAGS CREATION, STORAGE AND SEARCH RETRIEVAL, published as U.S. Patent Application Publication No. US 2010/0303288, now U.S. Patent No. 8,135,169, issued on March 13, 2012 (Atty. Dkt. No. MPOR-30156). U.S. Patent Application Publication No. 2010/0303288 and U.S. Patent No. 8,135,169 are incorporated by reference herein.

[0003] U.S. Patent Application No. 12/857,358 is a Continuation of U.S. Patent Application No. 11/621,062, filed on January 8, 2007, and entitled METHOD FOR MULTI-MEDIA RECOGNITION, DATA CONVERSION, CREATION OF METATAGS, STORAGE AND SEARCH RETRIEVAL, published as U.S. Patent Application Publication No. U.S. 2007/0150517, now U.S. Patent No. 7,778,438, issued on August 17, 2010 (Atty. Dkt. No. MPOR-28133). U.S. Patent Application Publication No. U.S. 2007/0150517 and U.S. Patent No. 7,778,438 are incorporated by reference herein.

[0004] U.S. Patent Application No. 11/621,062 is a Continuation-in-Part of U.S. Patent Application No. 11/325,373, filed January 4, 2006, and entitled APPARATUS FOR CAPTURING INFORMATION AS A FILE AND ENHANCING THE FILE WITH EMBEDDED INFORMATION, published as U.S. Patent Application Publication No. U.S. 2006/0115111, now U.S. Patent No. 7,184,573, issued on February 27, 2007 (Atty. Dkt. No. MPOR-27243). Application No. 11/621,062 claims benefit of U.S. Provisional Application No. 60/757,075, filed on January 6, 2006, and entitled APPARATUS AND METHOD FOR EMBEDDING META-TAGS INTO MEDIA FILES (Atty. Dkt. No. MPOR-27490). U.S. Patent Application Publication No. U.S. 2006/0115111 and U.S. Patent No. 7,184,573 are incorporated by reference herein.

[0005] U.S. Patent Application No. 11/325,373 is a Continuation of U.S. Patent Application No. 10/674,910, filed September 29, 2003, and entitled FORENSIC COMMUNICATION APPARATUS AND METHOD, published as U.S. Patent Application Publication No. U.S. 2004/0125208, now U.S. Patent No. 6,996,251, issued on February 7, 2006 (Atty. Dkt. No. MPOR-26491). Application No. 10/674,910 claims benefit of U.S. Provisional Application No.

60/414,449, filed September 30, 2002, and entitled FORENSIC COMMUNICATION APPARATUS AND METHOD. U.S. Patent Application Publication No. U.S. 2004/0125208, U.S. Patent No. 6,996,251, and U.S. Application No. 60/414,449 are incorporated by reference herein.

TECHNICAL FIELD

[0006] This invention relates to the storage and search retrieval of all types of digital media files, whether music or other audio, still photographs, videos, movies or other types of media.

BACKGROUND

[0007] The popularity of digital media devices such as digital cameras, video cameras, mobile phones with audio and video recording capability and portable music devices that have recording capability has exploded in recent years. Instead of recording pictures, video and sound on physical media, modern devices record to rewritable memory devices. This means that the cost to create a new image, movie or audio recording has fallen to near zero, making the number of these recordings available in any given individual's library skyrocket.

[0008] But this new ability to store virtually unlimited numbers of media files introduces new problems. First, the sheer number makes it nearly impossible to manually describe and index every media file in one's possession. This means that, for many, photographs, movies and audio recordings are kept in poorly organized computer files and finding any particular picture, movie or recording is a difficult and time-consuming process.

[0009] The second problem is the fact that people typically trade up to a new computer every three years or so. This means that hours of video, thousands of pictures or hundreds of audio files must be transferred from the old system to the new -- a sometimes daunting task.

[0010] A third problem is one can typically access locally stored media files only on the computer on which they reside. If one wishes to share the file with another one must typically employ some file-transfer method ahead of time (email, FTP, public server, etc.)

[0011] A fourth problem relates to e-mailing or sending your media files to another party, whereas the receiving party is not able to search the media files for the specific key indexes that the original owner had intended. While there are programs to allow the originator to type in key index words (tags) for searching and retrieving these media files from their personal computer, when these media files are e-mailed or sent to another party, these tags are removed from the media file, therefore the receiving party does not have an ability to search, sort, display, play or print these media files based on the original owners key indexes.

[0012] Finally, those who make a living providing content need some method for proving that a given work belongs to them, and that they are the original creator of the work.

SUMMARY

[0013] The present invention disclosed and claimed in one aspect thereof a system for capturing data in a first media and storing in a database at a remote location on a network. A network interface device is provided having a first capture device interfacing with a first external information source that generates external information and capturing the first external information during generation thereof. A processor is provided for processing the captured first external information and storing it in a first media format as stored first captured information within the network interface device, the processor initiating the storage of the first captured information as stored first captured information at an initial time and completes storage of the first captured information as stored first captured information at a completion time, thus providing a stored defined set of first captured information representing the first captured information between the initial time and the completion time. A transmitter is provided for transmitting as a transmitted defined set of first captured information the defined set of stored captured information to a remote location on a network after the completion time. A remote processing system is disposed at the remote node on the network and includes a database and a receiver for receiving the transmitted defined set of first captured information from the transmitter associated with the network interface device as a received defined set of first captured information. A data converter is operable to convert the received defined set of first captured information to a second format as a set of converted first captured information, the second format different than the first media format, the second format being a searchable format. The database then stores the set of converted captured information.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] For a more complete understanding of the present invention and the advantages thereof, reference is now made to the following description taken in conjunction with the accompanying Drawings in which:

[0015] Fig. 1 illustrates a block diagram of the overall operation of the system in accordance with the present disclosure.

[0016] Fig. 2 illustrates a block diagram of one embodiment of the overall operation of the system in accordance with the present disclosure.

[0017] Fig. 3 illustrates a block diagram of another embodiment of the overall operation of the system in accordance with the present disclosure.

[0018] Fig. 4 illustrates a block diagram of the file management of the system in accordance with the present disclosure.

DETAILED DESCRIPTION

[0019] Referring now to Fig. 1, there is illustrated a block diagram of the overall operation of the system in accordance with the present disclosure. The invention is best described by beginning with the capture device 100.

[0020] Still pictures, moving pictures, audio, telemetry or other information (hereafter called simply, "information") is gathered by the data converter 102 and organized into one "data element" 104 consisting of a single picture, a movie/video clip, an audio clip, a sample of telemetry data or other logical grouping of related information. The data converter 102 can be any type of data capture information. For pictures, a digital camera can be utilized and, for movie/video clips, the data converter can be a digital video recorder (DVR). In general, the data converter is any type of device that will capture the information and place it in some type of digitized format. As will be understood and discussed herein below, this digitized format is typically native to the data converter and the manufacturer. Some manufacturers have a particular file format in which they generate the information. This file format may have a designator as to the user, a sequence number or time stamp.

[0021] At the same time the data element is created by the capture device 100, certain other data ("meta data") 106 is captured as well. This meta data may include time and date 158, location 160, operator identification 162, or other information deemed relevant to establish the context of the data element. The time, date and location may be taken from a GPS unit, or may be derived by time-domain analysis of certain types of mobile phone signals or wireless network domains.

[0022] In some embodiments of this invention, it may be desirable to use a technique known as *steganography* to permanently and indelibly embed the meta data directly into the data element. By using one of several well-known steganographic techniques, the data element can be subtly altered such that 1) it is difficult to determine that hidden information is stored in the data element, and 2 even if it is known that hidden information is stored in the data element, retrieval without some secret information (a "key") is difficult. In this way, the data element can be guaranteed authentic even if stripped from external meta data.

[0023] Also, at the time of information capture, the capture device may gather additional information from the operator by means of a secondary data converter 108 that relates to defining the context of the data element. For example, after a camera / video recorder takes a picture / video, a microphone (the secondary data converter) might capture the audio describing the image or the audio from the video just captured. This "context description element" 110 is stored along with the data element and the meta data.

[0024] At this point, the capture device 100 has in its internal temporary storage the data element, the meta data and optionally the context description element. It now creates a composite data set using one of a number of well-known algorithms for combining multiple data sets into a single data set. For example, the well-known ZIP compression algorithm routinely performs statistical compression on a number of input files and creates a single output file such that, with the proper decompression algorithm, the original set of files can be recovered without data loss. This "combine and compress" function 112 creates a data set called the "composite element." 114.

[0025] The capture device now encrypts the composite element using any of a number of well-known encryption algorithms. In the preferred embodiment, the composite element is first encrypted using a symmetrical cryptosystem 116 using a key 118 shared by the capture device and the storage facility. The resulting "singly-encrypted element" 120 is then hashed. "Hashing" 122 is a technique that generates a digital "signature" for a data set, such that any party wishing to verify the correctness of the data set can easily recalculate the hash and compares it with the previously calculated hash. Hash algorithms have the further property that it is computationally difficult to find multiple data sets that have the same hash value. This calculated hash value ("element hash") 124 becomes part of the meta data set that is to be associated with the data element.

[0026] It is now necessary for the capture device to contact the storage facility over some communications medium. This communications medium 126 can be any mechanism that permits near real-time, two-way communication. The specifics of the communication medium are not disclosed here, but could comprise a wireless telephone network, the public switched telephone network, or the Internet. The capture device sends the meta data 106 (now consisting of the time,

date, operator identification, image, video, audio, "context description element" 110, hash 122 and possibly other data) to the storage facility 128.

[0027] The storage facility validates the meta data received from the capture device and, if authorized, combines and hashes 130 the received meta data and a secret number 132 known only to the storage facility. This hash is appended to the original meta data set and is then returned to the capture device as a "certificate." 134 At a later time, the certificate can be presented to the storage facility to determine if a data element is authentic.

[0028] The singly encrypted element 120 is now combined with the certificate 134 and then is further encrypted using an asymmetrical cryptosystem 136 under the public encryption key 164 of the storage facility 128, designated K_{SU}. This new packet (the "transport packet") 138 is now ready to transmit to the storage facility 128.

[0029] When it receives the transport packet 138, the storage facility 128 first decrypts 140 the packet 138 using its private decryption key 166 designated as K_{SR}. This gives the storage facility 128 access to the certificate 134 (which contains the meta data 106). The certificate can be validated 142 by rehashing the meta data 106 and the secret number 132. Unpacking the certificate 134, the storage facility 128 now knows the identity of the capture device 100, and can use this to retrieve the secret, symmetrical key 118 under which the singly encrypted element 120 is encrypted. The storage facility 128 now decrypts 149) the singly encrypted element 120 to retrieve the composite element 114; then expands and recovers 150 the composite element to retrieve the data element 104, the meta data 106, and the context description element 110, if present.

[0030] Now, the storage facility 128 can store the data element 104 under the owner's account in its mass storage 144. The storage facility knows the owner's identity because it was disclosed in the meta data 106. The storage facility also adds "context tags" 146 by methods dependent on the type of data that is being stored.

[0031] Context tags 146 are searchable elements derived from either the data element 104 itself or from the context description element 110. For example, if the data element 104 is a still

photograph or video, the storage facility may create context tags that describe elements of the scene or image(s), such as "animal," or "dog," or "Spot," depending on the mechanism that converts the information in the data element or the context description element into a tag.

[0032] It is equipment analyze the data elements (photograph, movie, audio recording, etc.) and create 148 a set of appropriate tags. For audio files, this may include a speech-to-text algorithm; for still or moving images, it may include image recognition and identification. Whatever the method used, at the end of the process the set of data to store includes the data element 102, the context element 110, and meta data 106 that now includes a set of searchable tags specific to that image, video, audio or other media. 146, presumed that, as image and voice recognition improve; this task can be fully automated. Therefore, the preferred embodiment of this invention is to have the task automated.

[0033] Retrieval of the data elements is performed from some access point 152 remote from the storage facility 128. To retrieve data elements, a client must prove to the storage facility that the client is authorized to access the desired data element. This can be done in any number of ways, but in the preferred embodiment of the invention, it is a password challenge. The access point 152 creates an authentication request 154 and transmits the request via the communications medium 126 to the storage facility 128. At the storage facility, the request is authenticated 156. Once the challenge has been successfully met, the client can access the data elements belonging to it. Details of the authentication process may include password authentication, a challenge/response protocol, or may employ a hardware authentication device.

[0034] Once the request for information has been authenticated, the storage facility 128 reads the requested information from the bulk storage device 144 and encrypts 169) under the public key of the requester 168, designated as K_{CU}. The encrypted data is then transmitted to the client over the communications medium 126. Upon arrival, the message is decrypted 172 under the client's private key 170 designated as K_{CR} and the data 174 is recovered.

[0035] Once the data has been retrieved, the tags may be edited or removed and new tags added. Other meta data; particularly location and time and date cannot be changed.

[0036] Variations of the system include placing the ability to enter tags on the data capture device itself. This could be in the form of a keypad, a touch screen or voice recognition software. If this option were taken, the data packet from the image capture device would include the tags in the meta data rather than a context description element.

[0037] Another variation applies to highly secure applications in which it is desirable to keep the data element 104 encrypted even while at the storage facility. In this variation, the data element 104 is encrypted under a symmetrical cryptosystem prior to combination with the meta data 106 and the context description element 110. This variation precludes the automatic extraction of tags from the data element itself, but still permits tagging based on the context description element.

[0038] Referring now to Fig. 2, there is illustrated a diagrammatic view of the overall operation of the system. In this system, the capture device 100 is disclosed wherein the actual device that captured it, the device 102, is illustrated as being a digital camera. This has a lens 202 associated therewith, with the output of the camera being a digitized image 204. This is basically the data element 104 of Fig. 1. It can be seen that the element 204 is in a digitized format that is typically native to the camera. This can be any type of video capture element for capturing an image 206.

[0039] Once the image is captured, it is stored in a data base 208. In addition to this information, various associated information such as audio, timestamp, GPS, location, etc. can be collected. One method for collecting the audio information, for example, is to utilize a microphone 210 that will capture the information and then digitize it in a digitizing block 212 utilizing an analog-to-digital converter, for example. This not only converts it to a digital value but, further, will convert it to a particular audio format such as a *.WAV file format. This particular file format is stored in the database 208. Of course, any other type of digital format could be utilized that is consistent with an audio format. The GPS information for location can be collected with an external GPS system 218 and timestamp information can internally be generated.

[0040] After all the information regarding the video information and the audio information, for example, is collected, it is stored in the database 208 and then must be output therefrom. In this embodiment, there are two types of attached information that are to be embedded within the image at a later time. The first set of information is the GPS information, the timestamp information, etc., that is collected automatically with any image. This information is created in a temporal relationship with respect to that particular image at the time of the capture of the image information. This location information, timestamp information, etc., is information that is unique to the photograph and defines that image. Further, user information can be provided which defines the user information that is associated with the capture device, i.e., the camera. The additional information, the audio information, is provided in the form of comments and the such which can be stored. Therefore, when the data in the form of the image information is to be transmitted to the remote site, it is combined with the additional GPS, location, timestamp, etc., information and the audio input information.

[0041] There is typically provided a unique file format that defines the digital image and this unique file name can be utilized to define all of the secondary information such that there is a unique association of that information with the image. Thereafter, a compress module 220 is provided for compressing the information in a compressed file format such as a *.ZIP file format. This is just a manner to transmit a number of files together. However, at the reception point, when the files are extracted from this *.ZIP file, there must be some way to distinguish the files and again associate them. This is done, for example, with a unique file naming structure. However, there could be other techniques utilized to uniquely identify the association between these different files.

[0042] Once this compressed file format is transmitted to the storage facility 128, it is stored in a database 226. At this time there will be, for example, a video clip or a video image (such as a still image) stored in association with the various information that is associated therewith. This, in effect, is a relational database that provides data storage in close association with each other. The first thing that must be done is to extract the information from the data. This is done in a block 228 wherein the associated information is extracted from the database, this being the associated information, and then processed. Typically, the associated information will be the

audio information in the audio file format. This must be converted. One conversion that is provided for is to convert the audio formatted data to text data. Therefore, one type of audio-to-text converter can be a voice translation system. There are many of these that are provided such as the Dragon Naturally Speaking systems.

[0043] Once the text format has been provided, this is a converted to intermediate formatted data, i.e., text data, that can then be processed in a format that can be embedded within a video file or an image file. This can then be converted into HTML data or other data. This will typically be formatted such that it can be defined as a meta tag for association with the video image. This meta tag is then combined with the image in a block 230. Once combined, this will then be stored in the database in association with the original raw video and raw audio files. Thereafter, there is an access and retrieval block 152 that can allow one to access the particular modified or "tagged" image via a search. There can be provided a search algorithm that searches all of the tagged images. This searching can be performed based upon the GPS location information, the timestamp information, the added audio comment information, etc. Any information that can be provided over and above the video information that was provided in all of the associated information at the camera can then be searched, as this is the information that is contained in the appended information to the image.

[0044] Referring now to Fig. 3, there is illustrated an alternate embodiment wherein substantially all of the combining operation is contained within the capture device 100 or the camera. Again, there is provided the capture device 102 in the form of the camera that captures the image 206. This is converted and stored in a database 306. The database 306 is basically the combination of database 208 and the database 226. Initially, all of the information from the digital video image 204 and the audio information and all other associated information such as the GPS information, timestamp, etc., are all stored in the database 306. There will be a corresponding process 308 for taking the associated information and converting it into different information, i.e., a meta tag, which is substantially the same as the process 228. Once the associated process is combined it is converted into that format, then it can be combined with the image in a process block 310, similar to the process block 230. Once this occurs, then there is provided a combined video/image with meta data that can be transmitted. This is illustrated as

the augmented image 316 which has meta data associated therewith. This is transmitted for storage at the storage facility in a database 318. This is searchable through the access/retrieval process 152, as described herein above.

[0045] Referring now to Fig. 4, there is illustrated one exemplary embodiment of the file structure. The image 204 is a digitized image that constitutes a particular file folder that will have a particular video or image format. This could be a JPEG format, an MPEG format or any other type of video format. This is referred to as one having the extension *.VID (for generic purposes). The file format in this embodiment will have a user ID section 402, a time/date stamp section 404, a sequence number 406 and a device number 408. This will have the format of XXX.AAAA.XXX.XXX.VID. The user ID section 402 will define the user of the capture device, this typically being configuration information that is input to the system. The time/date stamp is time and date information that can be taken from an internal clock or it can be derived from some external accurate time source. The sequence number 406 is typically internally generated with a counter that represents an internal sequence that is unique to a particular capture device. However, among different capture devices, the sequence could be the same. The device section 408 is a unique device number given to a particular device. Therefore, with the user ID information, the time/date stamp, the sequence number and the device number, a very unique file number will be generated. Further, the sequence and the time/date information will be different for each file. Thereafter, the information retrieved from the GPS 218 will be disposed in a folder with an extension of, for example, *.GPS. This will be stored in the file folder 208. The file numbering will be identical to the file nomenclature other than the extension will be identical to that of the digitized video file. Similarly, the audio information will be stored in a file folder with an extension of, for example, *.WAV with the body of the file name being identical to that of the digitized video file. This will all be stored in the database 208 and then combined in a compressed folder of the *.ZIP type. The actual file name for this can be any type of file name and it does not have to be identical or unique with respect to the name. However, there should be some type of unique file name in that, a random filename could be duplicated by other capture devices. Thus, in the preferred embodiment of this disclosure, the unique body of the *.ZIP file will be identical to that associated with the files contained therein such that this will have a filename of "XXX.AAAA.XXX.XXX.ZIP." This is the file that is transmitted.

[0046] As a summary, the system of the present disclosure provides a system for capturing, storing, indexing and retrieving data objects, which can include a capture device, a storage facility and an access point. The system consists of a primary data converter, a secondary data converter, a meta data source, a data combiner and compressor, a symmetrical encryptor, a one-way hash function, an asymmetric encryptor, and a communications port. The primary data converter captures some physical phenomenon such as, but not limited to a still image, a moving image, a sound, or some other factor, into a primary data set. The secondary data converter is capable of capturing some phenomenon into a secondary data set, related to but separate from the information captured by the primary data converter. The meta data source produces a device identifier, time, date, location, and other data related to the information captured by the primary data converter into a meta data set. The source of the time, date and location information is a GPS receiver, a wireless receiver or another receiver. The source of the device identifier is a read-only memory device. The data combiner and compressor is capable of combining the output of the multiple sources of data (the primary data converter, the secondary data converter, and the meta data source) into a single data stream, and then compressing said data stream into a compressed data set such that the compressed data set requires fewer transmission or storage resources than the uncompressed stream, but remains recoverable such that the original data from the primary data converter, the secondary data converter and the meta data source can be recovered without error. The symmetrical encryptor is capable of using a key shared with another party to convert the data stream from the data combiner and compressor into a singly encrypted data set that is unusable by any party other than the party that has knowledge of the key. The one-way hash function is capable of calculating for the encrypted data stream from the symmetric encryptor a number associated with said data stream such that (a) the number represents the data stream, but the data stream is not recoverable from the number, and (b) that it is computationally infeasible to created a second data stream that, when presented to the one-way hash function, produces an identical number. The communications port is capable of sending the meta data and the hash to a second party. The communications port is further capable of receiving from a second party a certificate that has the property of (a) being uniquely and verifiably identified with the meta data and hash of claim 12, and (b) being verifiably identified as originating with the second party. The asymmetric encryptor is capable of converting the output of the symmetric encryptor and other data into an encrypted information packet that can

be read only by a specific second party by means of a pair of related but non-identical keys, the encryption key and the decryption key. The communications port is further capable of conveying the encrypted information packet to a second party. The storage facility consists of a communications port, a device authenticator, an asymmetric decryptor, a validator, a symmetric decryptor, a data expander and recovery device, a tag generator, a mass storage mechanism, an asymmetric encryptor, and a user authenticator. The communications port is capable of receiving a request from validation from the capture device. The device authenticator is capable of verifying that the capture device is authorized to use the storage facility and to create an authentication certificate such that (a) it is computationally infeasible to create a second meta data set or hash that creates an identical authentication record, and (b) the authentication record is uniquely identified with the storage facility. The asymmetric encryptor is capable of using the decryption key to recover the authentication certificate and the singly encrypted data set. The validator is capable of determining if the recovered authentication certificate (a) was generated by the storage facility, and (b) is valid for the presented hash and meta data. The symmetric decryptor is capable of converting the singly encrypted data set into the compressed data set. The data expander and recovery device is capable of converting the compressed data set into the original primary data set, the secondary data set, and the meta data set. The tag generator is capable of taking the primary data set, the secondary data set and the meta data set and producing a set of index tags that describe the primary data set. The tag generator in which the tag generation is performed by a human operator. The tag generation is performed by a speech-to-text function or by an image recognizer. The mass storage mechanism is capable of storing the primary data set, the secondary data set and the meta data set in a way that the information can be retrieved based on the index tags. The user authenticator is capable of receiving requests from access points and verifying their authority to perform retrieval operations at the storage facility.

[0047] The system can, at its most condensed version, comprise an asymmetric encryptor capable of converting data read from the mass storage mechanism using a public encryption key into a form usable only by a party with knowledge of a secret key that corresponds to the said public encryption key. The access point consists of an asymmetric decryptor, a communications port and an authentication requester. The authentication requester is capable of identifying the

access point and the data element or elements to be recovered from the storage facility in a manner that proves its authority to access said data element or elements.

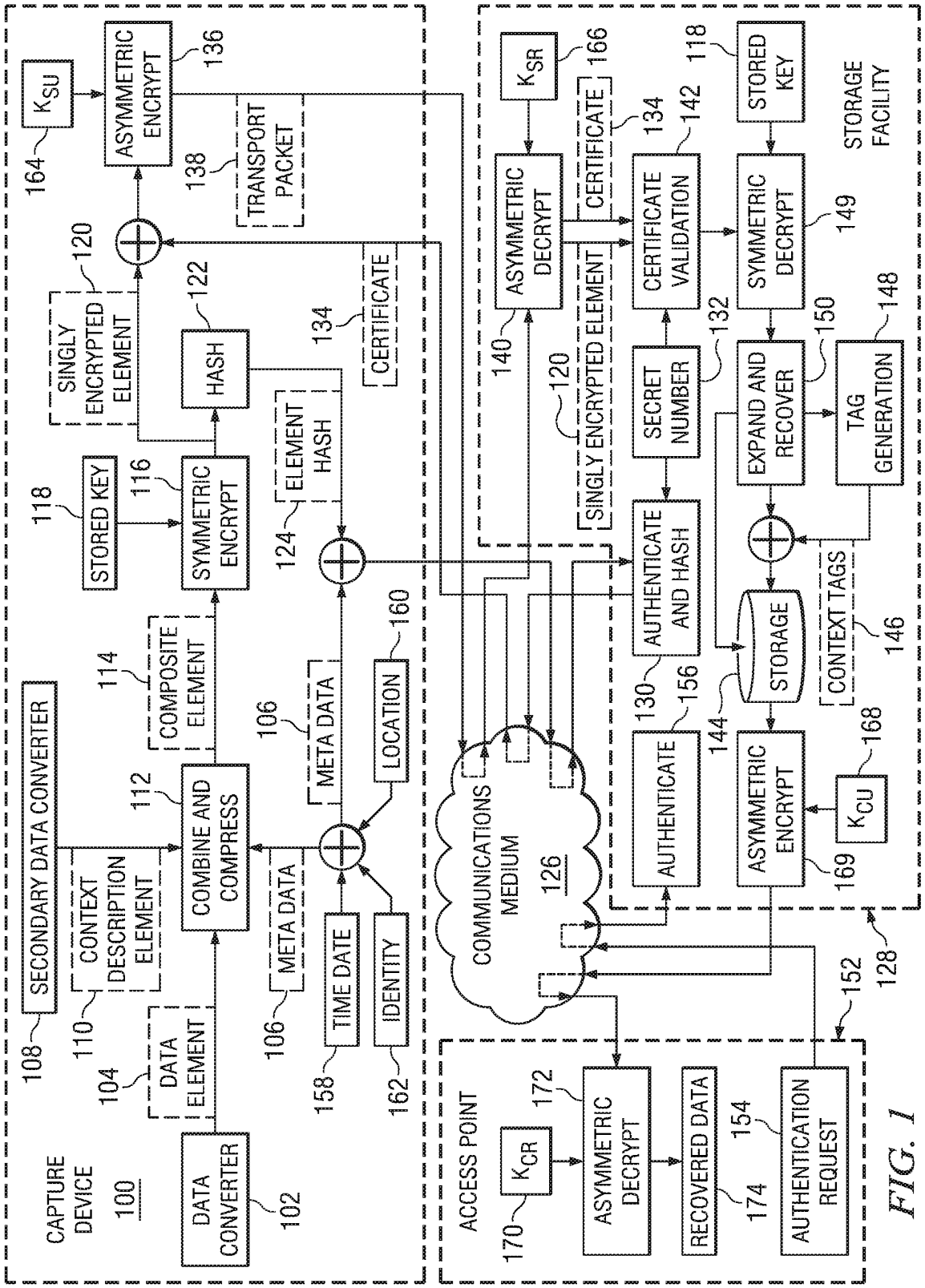

CAPTURE DEVICE
100
DATA CONVERTER
102
DATA ELEMENT
104
SECONDARY DATA CONVERTER
108
CONTEXT DESCRIPTION ELEMENT
110
COMBINE AND COMPRESS
112
COMPOSITE ELEMENT
114
META DATA
106
TIME DATE
158
IDENTITY
162
LOCATION
160
STORED KEY
118
SYMMETRIC ENCRYPT
116
SINGLY ENCRYPTED ELEMENT
120
HASH
122
ELEMENT HASH
124
CERTIFICATE
134
ASYMMETRIC ENCRYPT
136
TRANSPORT PACKET
138
K_{SU}
164
ACCESS POINT
K_{CR}
170
ASYMMETRIC DECRYPT
172
RECOVERED DATA
174
AUTHENTICATION REQUEST
154
COMMUNICATIONS MEDIUM
126
152
128
AUTHENTICATE
130
AUTHENTICATE AND HASH
156
SECRET NUMBER
132
ASYMMETRIC ENCRYPT
144
K_{CU}
168
STORAGE
169
EXPAND AND RECOVER
150
TAG GENERATION
148
CONTEXT TAGS
146
ASYMMETRIC DECRYPT
140
SINGLY ENCRYPTED ELEMENT
120
K_{SR}
166
CERTIFICATE
134
CERTIFICATE VALIDATION
142
SYMMETRIC DECRYPT
149
STORED KEY
118
STORAGE FACILITY

FIG. 1

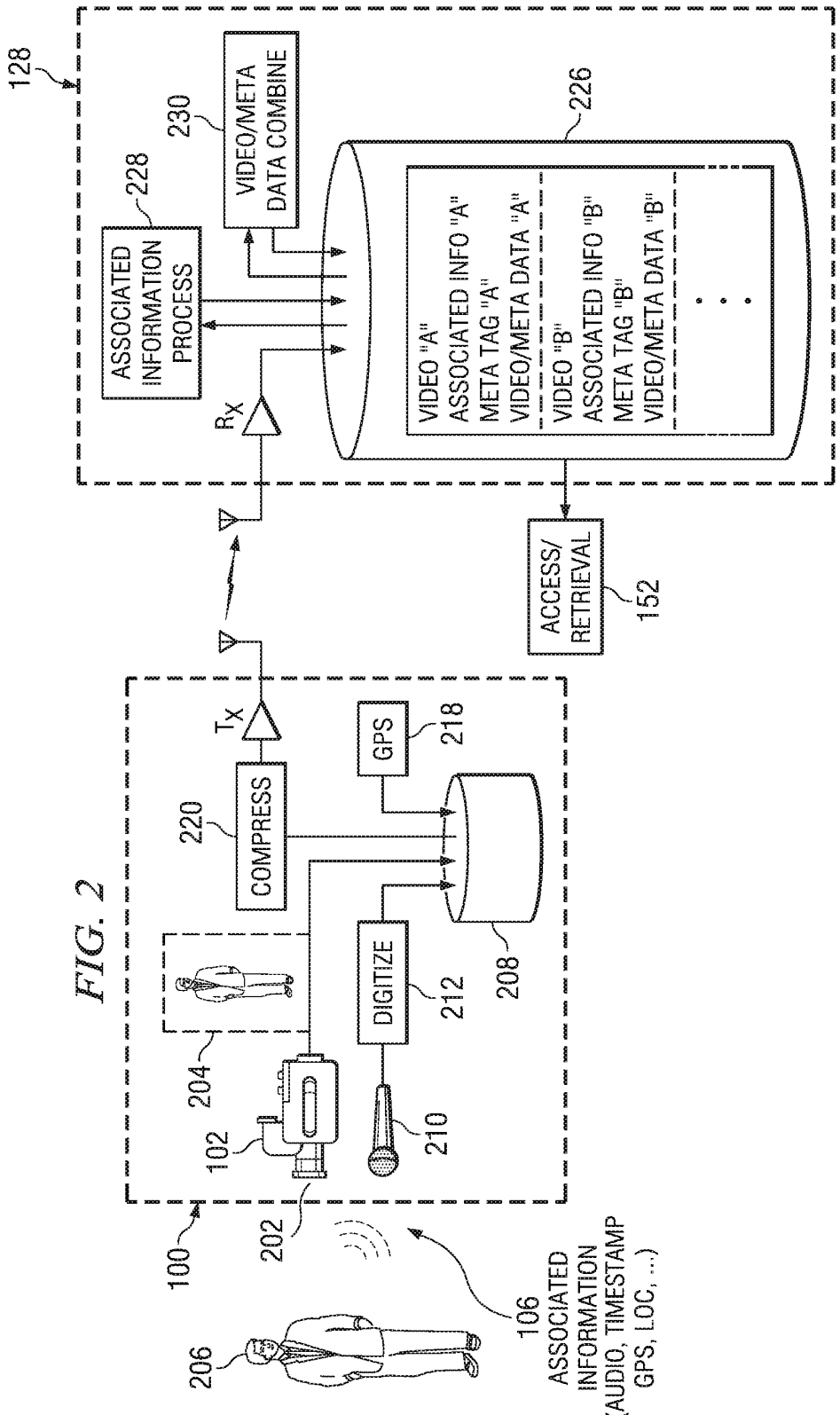

128
230
VIDEO/META DATA COMBINE
228
ASSOCIATED INFORMATION PROCESS
226
VIDEO "A"
ASSOCIATED INFO "A"
META TAG "A"
VIDEO/META DATA "A"
VIDEO "B"
ASSOCIATED INFO "B"
META TAG "B"
VIDEO/META DATA "B"
Rx
ACCESS/ RETRIEVAL
152
Tx
GPS
218
220
COMPRESS
204
102
DIGITIZE
212
208
210
100
202
206
106
ASSOCIATED INFORMATION (AUDIO, TIMESTAMP GPS, LOC, ...)

FIG. 2

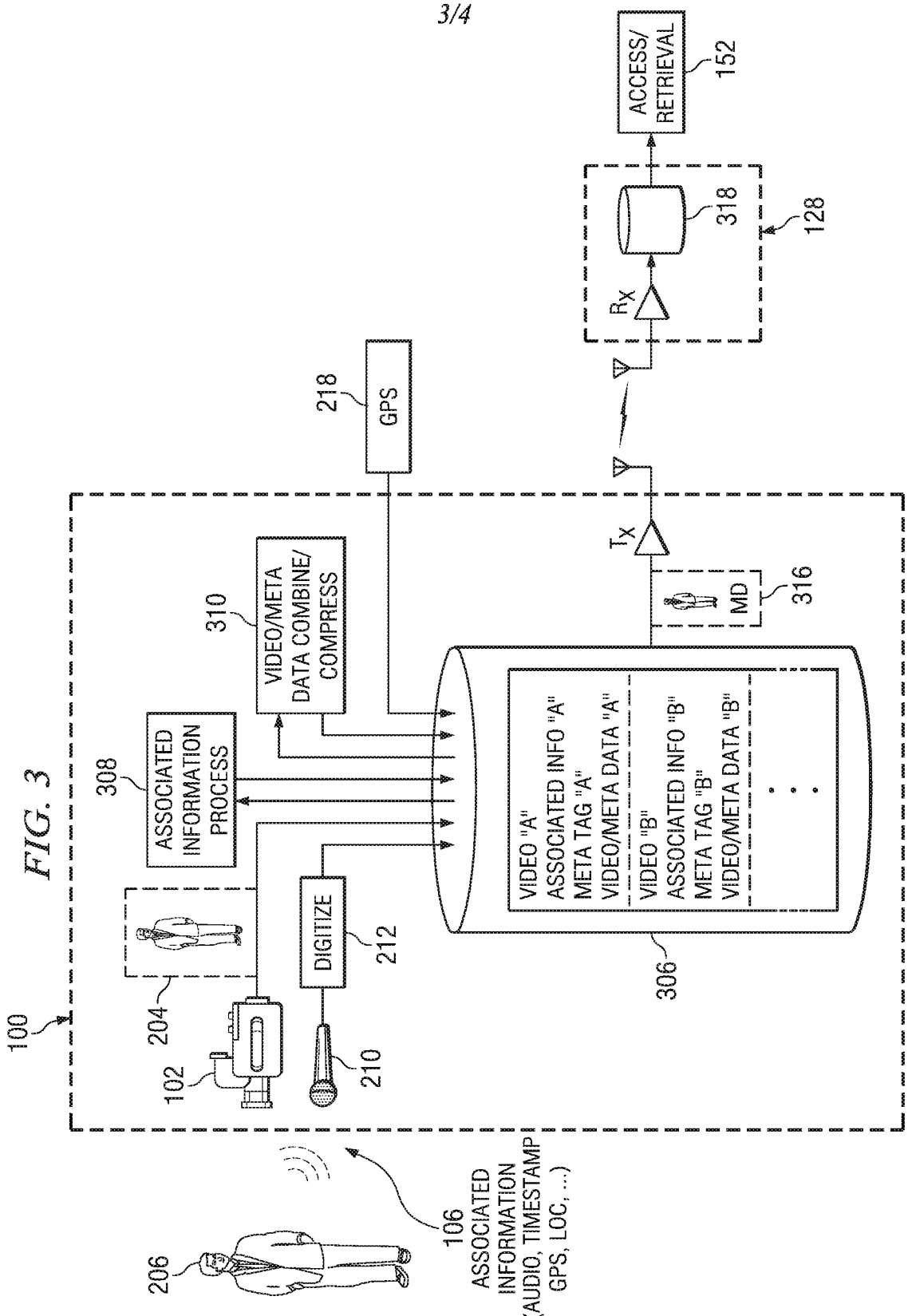

100
102
204
206
210
212
218
GPS
DIGITIZE
ASSOCIATED INFORMATION PROCESS
308
VIDEO/META DATA COMBINE/ COMPRESS
310
106 ASSOCIATED INFORMATION (AUDIO, TIMESTAMP GPS, LOC, ...)
VIDEO "A" ASSOCIATED INFO "A" META TAG "A" VIDEO/META DATA "A"
VIDEO "B" ASSOCIATED INFO "B" META TAG "B" VIDEO/META DATA "B"
306
MD
316
Tx
Rx
318
128
ACCESS/ RETRIEVAL
152

FIG. 3

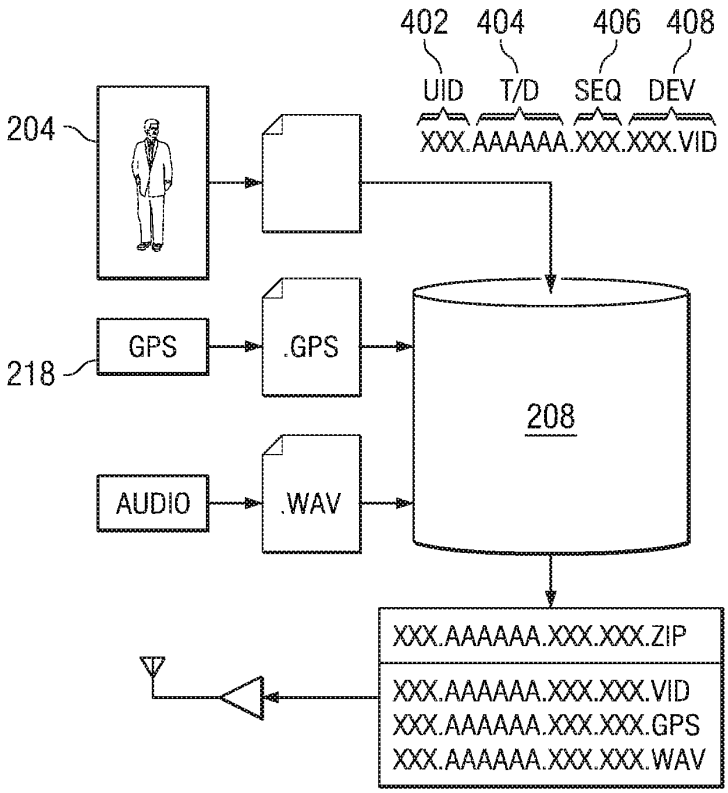

402
404
406
408
UID
T/D
SEQ
DEV
XXX.AAAAAA.XXX.XXX.VID
204
GPS
.GPS
218
208
AUDIO
.WAV
XXX.AAAAAA.XXX.XXX.ZIP
XXX.AAAAAA.XXX.XXX.VID
XXX.AAAAAA.XXX.XXX.GPS
XXX.AAAAAA.XXX.XXX.WAV

FIG. 4