Doc Code: MES.GIB
Document Description: Certification of Micro Entity Status (Gross Income Basis)

PTO/SB/15A (03-13)

CERTIFICATION OF MICRO ENTITY STATUS
(GROSS INCOME BASIS)

Application Number or Control Number (if applicable):	Patent Number (if applicable):
First Named Inventor: KREBS	Title of Invention: MULTIMEDIA SYSTEM FOR MOBILE CLIENT PLATFORMS

The applicant hereby certifies the following—

(1) **SMALL ENTITY REQUIREMENT** - The applicant qualifies as a small entity as defined in 37 CFR 1.27.

(2) **APPLICATION FILING LIMIT** - Neither the applicant nor the inventor nor a joint inventor has been named as the inventor or a joint inventor on more than four previously filed U.S. patent applications, excluding provisional applications and international applications under the Patent Cooperation Treaty (PCT) for which the basic national fee under 37 CFR 1.492(a) was not paid, and also excluding patent applications for which the applicant has assigned all ownership rights or is obligated to assign all ownership rights as a result of the applicant's previous employment.

(3) **GROSS INCOME LIMIT ON APPLICANTS AND INVENTORS** - Neither the applicant nor the inventor nor a joint inventor, in the calendar year preceding the calendar year in which the applicable fee is being paid, had a gross income, as defined in section 61(a) of the Internal Revenue Code of 1986 (26 U.S.C. 61(a)), exceeding the "Maximum Qualifying Gross Income" reported on the USPTO website at http://www.uspto.gov/patents/law/micro_entity.jsp which is equal to three times the median household income for that preceding calendar year, as most recently reported by the Bureau of the Census.

(4) **GROSS INCOME LIMIT ON PARTIES WITH AN "OWNERSHIP INTEREST"** - Neither the applicant nor the inventor nor a joint inventor has assigned, granted, or conveyed, nor is under an obligation by contract or law to assign, grant, or convey, a license or other ownership interest in the application concerned to an entity that, in the calendar year preceding the calendar year in which the applicable fee is being paid, had a gross income, as defined in section 61(a) of the Internal Revenue Code of 1986, exceeding the "Maximum Qualifying Gross Income" reported on the USPTO website at http://www.uspto.gov/patents/law/micro_entity.jsp which is equal to three times the median household income for that preceding calendar year, as most recently reported by the Bureau of the Census.

SIGNATURE by a party set forth in 37 CFR 1.33(b)

Signature	/Yasin Bismilla/				
Name	Yasin Bismilla				
Date	February 5, 2016	Telephone	647.680.3526	Registration No.	64756

☐ There is more than one inventor and I am one of the inventors who are jointly identified as the applicant. Additional certification form(s) signed by the other joint inventor(s) are included with this form.

MULTIMEDIA SYSTEM FOR MOBILE CLIENT PLATFORMS

FIELD OF INVENTION

The invention relates to issues of the wireless Internet, specifically to methods of multimedia transmission and playback for mobile clients.

BACKGROUND OF THE INVENTION

The current methods of distributing multimedia data over the wireless Internet to mobile clients are constrained by existing wireless bandwidth, and the real-time decoding, processing and displaying of multimedia content with limited hardware capabilities. These hardware limitations include slow CPUs, high memory latencies, slow drawing capabilities and the absence of YUV to RGB conversion in the hardware.

Video and audio playback exist on certain cell phone handsets, but this technology is embedded and takes advantage of low-level hardware processing to enable the performance required for media playback. Through embedded media players, several cell phone handsets and handheld computers can play either streamed video or audio. Popular digital video encoding standards for some handsets are H263 and MPEG4. The audio codecs, MP3, AMR and AAC, are also typically supported on some mobile handhelds. Newer video codecs, like H264 could be used for video transmission to cell phones, but would require

Google Exhibit No. 1008
Page 002

clients systems with fast memory access for their motion compensation methods.

Embedded streaming media players rely on firmware integration to take advantage of the multitasking capabilities of cell phone handsets. At the time of this writing, most cell phones cannot support multimedia playback because they are only capable of supporting one or a few concurrent processing threads. On handsets that have embedded media players, video is also limited to very low frame rates and the bandwidth available for streaming in North America is low, varying from 2-3kybtes/second to ISDN speeds of 64Kbits. European countries and Japan currently offer 3G network connection speeds, varying from 64kbits-300Kbits, and offer more technologically advanced cell phones with embedded media players that can achieve higher video frame rates. For limited per usage periods and some, EV-DO (Evolution Data Optimized) networks can also provide these higher speeds over local CDMA networks.

Decoders for complex video codecs which support highly scalable MPEG4 video, and more complex, CD quality music, audio codecs like AAC, require multiple parallel processes and fast processing. Mathematical algorithms, as in Fan and Madisetti, designed for the high number of floating point samples for higher end MPEG4 and AAC,

Google Exhibit No. 1008
Page 003

which requires a sample rate approximately 36,000 floating point calculations/second, are intended to run on specialized chips. Even at lower and very low bitrates, where MPEG4 is more efficient than its predecessors, MPEG4 software players depend on PC multitasking or hardware APIs for efficient processing to draw video frames.

Currently, device independent Java (J2ME) offers two standard configurations on mobile clients. The Connected Limited Device Configuration (CLDC) is prevalent in the J2ME world, and powers cellular phones, pagers, PDAs, and other handheld devices. A variant of CLDC, Connected Device Configuration (CDC) targets more powerful devices, such as home appliances, set-top boxes, and Internet TVs. The second configuration, Mobile Information Device Profile (MIDP), runs on top of the CLDC, and several profiles run on top of CDC.

Java players for cell phones like the Oplayo MVQ player exist, but implementations of true, platform independent, MPEG4 Java decoders that will play video on cell phones are not known. More efficient methods, such as U.S. Pat. No. 5,699,121, do not rely on DCT motion compensation, and propose pattern marching to identify regions in the motion residual signal that have not been accurately reproduced and to correct them using a pattern library. But again, this approach does not use MPEG4 video encoding.

Google Exhibit No. 1008
Page 004

Similarly, although Java decoders exist that play MP3 ringtones on cell phones, no Java players are known that will play AAC. In fact, many of the newer IDCT algorithms are targeted more towards customized logic chips that only do IDCT (composed of many simple pipelined instructions as opposed to a few more complex ones).

Bit streaming is the standard method of transmitting audio or video to cell phones over wireless networks. Streamed bits are buffered and then decoded, or entire video files are downloaded or proportionately cached, or as in progressive, http streaming downloading to a point where complete, continuous playback is deemed possible. In the case of wireless networks, bit streaming in terms of audio/video content is usually is done over a non-reliable transport like UDP and requires a lot of error correction and duplication of content (extra stream correction data). Advances in these transmission methods do propose more sophisticated means of reserved bandwidth, such as AROMA. Alternatives to streaming methods have been proposed for the transmission of video as objects through pre-fetched lists Waese et al., US Pat. No. 6,286,031,which are similar to downloading pre-fetched lists of SMIL objects, and instant or scheduled notification file downloading Stumm, US Pat. No. 5,768,528. However, these do not address specific continuity and deployment issues for wireless

Google Exhibit No. 1008
Page 005

multimedia transmission and concurrent playback on a limited-tasking cell phone handsets.

Streaming also requires client processing for significant error correction in video decoding, adaptive encoding for varying channel bitrates and in cellular networks. It also requires the use of cellular MMS multimedia protocol.

The object of the current invention is to solve one or more of the drawbacks in existing methods discussed above, or to provide other improvements to the art.

SUMMARY OF THE INVENTION

The invention relates to wireless Internet multimedia transmission and wireless clients. In a first aspect the invention provides a method of efficient multimedia object creation. In a second aspect the invention deployment addresses the limitations of large-scale multimedia transmission on cellular networks to wireless clients. In a third aspect, the invention relates to methods of decoding of video, sufficiently optimized to be played on a limited wireless client. In a fourth aspect, the invention relates to methods of decoding of audio, sufficiently optimized to be played on a limited wireless client. In a fifth aspect, the limited mobile handset multimedia object player, for both MPEG4

Google Exhibit No. 1008
Page 006

video decoding and AAC audio decoding, is implemented as a device-hardware-independent java (J2ME) applet.

In its first aspect, the invention pertains to efficiently transmittable multimedia object creation. A server-based transcoder, coupled with a multimedia object creator, inputs a standard analog signal or alternative digital signal like MPEG2, and converts this signal into true MPEG4/AAC multimedia objects. As multimedia objects they can then be dynamically uploaded to multiple live content hosting web servers, which, through proximate mobile network proxy servers, make live content accessible to mobile clients as consecutive multimedia objects.

The multimedia object creator produces discrete multimedia objects from video and audio segments of a continuous stream. If the stream is MPEG4, multimedia objects can also be segments of multiple component video and audio streams. In the case of multiple MPEG4 component streams, per object segmentation and decoding can enable the composition of a single scene from several temporally-independent multimedia objects. This provides the possibility of decoding only a limited number of multimedia objects, and not all objects, to provide an object-based scalability.

Multimedia objects are discrete, and also have distinctive Internet

Google Exhibit No. 1008
Page 007

addresses, and hence, the mobile client will have the opportunity to interact with any given media sequence on a per object basis.
In cases where the multimedia object sequence is being transcoded from a live stream, a window of multimedia objects is made available on the host server. This window would be comprised of a number of recently created multimedia objects. To minimize delays that will occur to maintain synchronicity between the client and server, a larger multimedia object window can be dynamically created on the host server.

In cases where the media object sequence has been previously transcoded and resides on the host as a non-live source, each object is numerically ordered. The transport mechanism for the multimedia objects is assumed to be HTTP for the purposes of illustration, however, other protocols which access content through file and directory structures could be used. For example FTP, IMAP4 and NNTP all have the capability to serve files in a directory structure.

On the client side, the number of multimedia objects that can be buffered in memory is based on the size of the first multimedia object The amount of free memory available and the processing of further multimedia objects in the sequence can be optional and dependent on

Google Exhibit No. 1008
Page 008

whether the implementation allows the modification of object parameters between multimedia objects (such as the size of the visual frame or the sample rate of the audio stream). The buffering and playback of multimedia objects in a synchronized fashion is critical to fluid playback.

The wireless networks over which the limited devices operate often have a very high latency. The HTTP 1.1 protocol and some implementations of HTTP 1.0 allow the use of a persistent connection over TCP to perform multiple requests. Furthermore, some HTTP 1.1 implementations allow the use of pipelined connections allowing the HTTP client to perform many requests in rapid succession decreasing the latency between the request and reception of each multimedia object. When possible, the invention can take advantage of this capability.

In its second aspect, the invention pertains to multimedia object deployment to large numbers widely distributed wireless Internet clients. Media content converted to multimedia objects must be available to many users and the distribution system must be sufficiently robust to allow peaks in demand and have sufficient geographic proximity that network congestion and latency are reduced.

Google Exhibit No. 1008
Page 009

In this second aspect, and in the case of live content transcoding from a live audio/video stream, the transcoding of input media formats and the creation multimedia objects is done in real-time and immediately deployed to every content server of the distributed system. These content servers may be at the same location, or they may be geographically placed to support local mobile clients and take advantage of alternative mobile network proxy servers and proxy server object caching. The distribution of multimedia objects to alternative content servers can take place on the wired Internet.

In a third aspect, the invention provides novel optimizations for digital video decoding. Some of these optimizations can then be used by an expert assessment process, whereby, the decoder maintains a state information list of short-cuts, related to perceived frame rate in a sorted list starting with those that will decrease output quality the least, to those that will decrease output quality the most but have the most impact on decoding speed. The client player-decoder dynamically adjusts how many shortcuts must be taken. These short-cuts are specifically designed to drastically reduce the number of computations necessary at certain critical steps in the video decoding process at the cost video output quality. This allows the video decoder to scale in complexity based on the processing power of the device

Google Exhibit No. 1008
Page 010

being used. It also allows users to experience multimedia playback despite the limitations of the device they may be using.

In a fourth aspect, the invention pertains to decoding audio on a limited mobile device. Mobile devices present many challenges for audio playback. Typical mobile processors have integer math only, little or no on-CPU cache, and a limited audio output interface. The present invention takes several approaches to getting maximum audio quality out of these very limited devices, which are applicable to other audio codecs such as AC3, AMR and WMA v9, as well as AAC LC. These include a novel use of Huffman codebooks, a highly optimized IMDCT process, and innovative windowing optimizations.

One of the serious limitations of mobile devices is their inability to play continuous sound. There is no way to play a long sound without gaps which occur when switching from one block of sound to the next block of sound. The proposed invention also solves this gapping problem by intelligent placement of the gaps. A frame of low total energy is selected, and the playback is controlled so that the gap will occur during that frame. The low-energy frame may be dropped so that synchronization is not lost.

In a fifth aspect, the invention pertains to the implementation of a

Google Exhibit No. 1008
Page 011

moble handset MPEG4 video and AAC audio player that is hardware-independent and operating system independent, and can simply be downloaded prior to media playback on mobile clients that do not have embedded media players. Hardware and operating system independence are characteristics of Java applets, but Java cannot take advantage of hardware capabilities in processing the huge number of calculations and variables required for either MPEG4 decoding or AAC decoding on limited processing mobile handset. Hence, the required optimizations for Java itself, to permit the playback of AAC and MPEG4 on current mobile client hardware, are a source of technological innovation and advance.

Google Exhibit No. 1008
Page 012

BRIEF DESCRIPTION OF THE DRAWINGS

For a better understanding of the present invention and to show more clearly how it may be carried into effect, reference will now be made, by way of example, to the accompanying drawings which show the preferred embodiment of the present invention and in which:

FIG. 1 is a general diagram for a distributed network system for multimedia-on-demand, utilizing a centralized content server, indexing host, multimedia object creator and transcoder for live broadcast applications or to transcode and create multimedia objects from archived multimedia files, and distributed content servers involving high capacity cellular network proxy servers and mobile clients running downloaded java applets or embedded or downloaded non-java multimedia object players; and

FIG. 2 is a flow diagram illustrating a multimedia object identification method by the multimedia object creator of FIG. 1 for mobile clients by the host content server of FIG. 1; and

FIG. 3 illustrates a multimedia object windowing sequence for a live transmission of multimedia objects created by the multimedia object creator of FIG. 1; and

Google Exhibit No. 1008
Page 013

FIG. 3a illustrates multimedia object creation for single stream multimedia, just audio and multi-stream MPEG4 composite layers by the multimedia object creator of FIG. 1; and

FIG. 4 is a flow diagram illustrating the steps of multimedia object processing by the multimedia object players of FIG. 1; and

FIG.5 is a diagram illustrating the architecture and processing interaction for the large scale distribution of live and archived multimedia content in the distributed network being managed by the indexing host of FIG. 1, involving remote transcoding/multimedia object creating servers and a central indexing host server; and

FIG. 6 is general diagram illustrating standard MPEG4 Simple Profile decoding steps which are followed in general for video decoding by the multimedia object players of FIG. 1; and

FIG. 7 is a flow diagram illustrating an optimized Huffman codebook method for digital video decoding method used by the multimedia object players of FIG. 1; and

FIG. 8 is a flow diagram illustrating a method of using a texture buffer to process P-frames for digital video decoding; and

Google Exhibit No. 1008
Page 014

FIG. 9 is a flow diagram showing a method of video decoding performing faster motion compensation without bilinear interpolation when less quality but faster processing is required that is used by the multimedia object players of FIG. 1; and

FIG. 10 is a flow diagram illustrating an optimized digital video decoding method for optimizations in pixel processing and dequantization used by the multimedia object players of FIG. 1; and

FIG. 11 is a flow diagram illustrating a novel use of Chen's algorithm used by the multimedia object players of FIG. 1; and

FIG. 12 is a flow diagram showing a novel handling YUV to RGB conversion used by the multimedia object players of FIG. 1; and

FIG. 13 is a flow diagram illustrating decoding short cuts for effective video decoding on variable limited mobile client hardware used by the multimedia object players of FIG. 1; and

FIG. 14 is a general diagram illustrating basic steps of the AAC digital audio decoding and other similar audio codec decoding, which are followed in general by the multimedia object players of FIG. 1; and

FIG. 15 is a flow diagram illustrating an optimized Huffman codebook method for digital audio decoding used by the multimedia object

Google Exhibit No. 1008
Page 015

players of FIG. 1; and

FIG. 16 is a flow diagram illustrating an optimized digital audio decoding method for optimizations in the IMDCT step used by the multimedia object players of FIG. 1; and

FIG. 17 illustrates simplified input short-cut processes specific to AAC Low Complexity (LC) audio decoding profile used by the multimedia object players of FIG. 1; and

FIG 18 shows audio decoding using an alternative bit-operation based Taylor computation method used by the multimedia object players of FIG. 1; and

FIG. 19 illustrates further IMDCT short window processing for digital audio decoding for the method used by the multimedia object players of FIG. 1; and

FIG. 20 illustrates low energy gap timing in audio playback for the method of audio decoding used by the multimedia object players of FIG. 1.

Google Exhibit No. 1008
Page 016

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Fig. 1 illustrates a centralized content server system 1, utilizing a transcoder 2 and a multimedia object creator 3 to create multimedia objects from a live broadcast 4 or to transcode and create multimedia objects from archived multimedia files 5. The central server includes an indexing host system 6 to deploy created multimedia objects to relevant content servers 7 through the wired Internet and to verify all geographically dispersed wireless clients 8. The system includes the potential use of proxy cellular network http servers 9, which can cache large numbers of small multimedia objects to support large numbers of concurrent wireless clients 8 running multimedia object java applets 10 or embedded or downloaded non-java multimedia players 11.

FIG. 2 is a flow diagram illustrating the process of multimedia object identification by the multimedia object creator 3. This process encodes a Supplied Identification to each multimedia object to identify the transport protocol, source host, path and number of objects of a particular multimedia stream.

The host directory name of the multimedia objects is formatted to contain the number of video objects located within the directory. An delimiting character is placed between the end of the directory name and the number indicating the multimedia object count. This allows

Google Exhibit No. 1008
Page 017

the use of directory names terminating in numbers while indicating an unambiguous multimedia object count e.g. StarWars_1.mp4, StarWars_2.mp4, etc.

Multimedia objects within the directory are named similarly to directory name. However, instead of the multimedia count following the delimiting character, a number indicating the multimedia object's position within the sequence of multimedia object is specified.
The following is an example:

Supplied Identification for MultimediaObjects:

<transport>://<host>/<path>/<MOSName><Delim><MOCount>

Computed Identification for each multimedia object based on Supplied Identification:

<SuppliedID>/<MOSName>.<MOSeqNum><Delim><MOType>

<transport> : protocol used to transmit multimedia objects

<host> : Host content servers 1 or 7 serving the multimedia objects directly to mobile clients 8

<path> : Path to the multimedia object directory

<MOSName> : Name of the multimedia object sequence (perhaps the name of a broadcast)

<MOCount> : Number of multimedia objects (Integer greater

Google Exhibit No. 1008
Page 018

than zero)

<MOSeqNum> : The multimedia object's sequence number

(Integer greater then zero, less then or equal to MOCount)

<Delim> : The delimiting character

<MOType> : enocoding type e.g. mp4

When multimedia has been transcoded from a non-live source, the first multimedia object in the sequence to be played could have the index 0 of a counting series. In cases where the multimedia object sequence is being transcoded by the multimedia object creator 3 from a live stream, a window of multimedia objects is made available on all content servers 1 and 7. This window would be comprised of a number of recently created multimedia objects transcoded from the live stream.

The window of multimedia objects allows clients to begin reception of a multimedia object sequence at an earlier point than the most recently created multimedia object. This mechanism provides an extra degree of forgiveness in high-latency situations, where there may be a delay between the client 8 discovering the most recent multimedia object and the actual request.

The window of multimedia objects would shift as more multimedia objects are transmitted from the live source. The multimedia object

Google Exhibit No. 1008
Page 019

sequences would begin at 0 and be numbered sequentially. The window size hence permits the removal of earlier objects.

For example, a live stream may be made comprised of a window of four objects. Upon transmission of a fifth video object, the first multimedia object would be deleted, resulting in the following sequence illustrated in FIG. 3.

The wireless client 8 can have the capability to search forward in the multimedia object sequence among the multimedia video objects in window. This provides additional transmission continuity in cases where it is not possible to maintain sufficient bandwidth for all multimedia objects in the live sequence.

To reduce delays that will occur to maintain synchronicity between the client 8 and server 1 or 7, a larger multimedia object window can be used . Likewise, the mobile client 8 may also store more than two multimedia objects in the internal buffer. Moreover, wireless networks over which the limited devices operate often have a very high latency. This is especially evident when TCP's 3-way handshake must be performed for every connection that is made. It is therefore ideal to use an application protocol that is able to minimize the latency between each request for a multimedia object.

Google Exhibit No. 1008
Page 020

The transport mechanism for multimedia objects is assumed to be HTTP for the purposes of system FIG. 1, however, other protocols which access content through file and directory structures could be used. For example FTP, IMAP4 and NNTP all have the capability to serve files in a directory structure. HTTP 1.1 protocol and some implementations of HTTP 1.0 allow the use of a persistent connection over TCP to perform multiple requests. Furthermore, some HTTP 1.1 implementations allow the use of pipelined connections, allowing the HTTP client to perform many requests in rapid succession decreasing the latency between the request and reception of each multimedia object.

In FIG. 3a, the transcoder 2 and multimedia object creator 3 create multimedia objects of an optimal digital encoding such as MPEG4/AAC from analog multimedia or an alternative codec stream 12, such as MPEG1, MPEG2, MOV, AVI, WMV, ASF, and higher encoded MPEG4. First, the input stream is transcoded into MPEG4 and AAC and then it is split according to a specified interval, such as 10 seconds, into multimedia objects. The video component of the stream is scanned after the specified interval for the next I-frame 13, where the split is made. Since typically there are no predicted frames in digitized audio 15, a conditional split is made to correspond to the video segmentation

Google Exhibit No. 1008
Page 021

14. In the case of multiple MPEG4 component streams 16, multiple video and/or audio composite layers can also be split into multimedia objects at I-frames 13. In the case of just audio signal input 12a, the transcoded audio can be analog, or digital codecs, AMR, MP3, RealAudio or higher rate encoded AAC and then split into specified intervals 13a.

FIG. 4 is a flow diagram illustrating client side processing of multimedia objects. Multimedia object player 10 or 11 processing is initiated by the receipt of the first multimedia object from a content server 1 or 7. The first multimedia object's Identification is parsed and the total number of multimedia objects stored within the Indentication's <path> is determined, or the case of live transmission applications, the number of multimedia objects in window. Hence, heap memory allocations for the multimedia objects and meta-data can then be determined. These allocations are created of sufficient size that multimedia objects that follow can verwrite older multimedia objects in the same memory allocation without overflowing.

Depending on the amount of heap memory available on the device, several sets of memory allocations can be made to store multiple multimedia objects at once. This constitutes a multimedia object buffer and allows the decoder to be playing one multimedia object

Google Exhibit No. 1008
Page 022

while the next has not yet fully completed reception. A device must have enough memory to allow two multimedia objects to be stored in the heap at once, otherwise the behavior is undefined (the decoder may refuse to play the multimedia object sequence, or play it with extended pauses between each multimedia object). Important state information is stored in several variables. This information includes the Integer values:

nMObjectPlaying – current Multimedia Object index playing

nMObjectRecving – current Multimedia Object index being received

and Boolean value(s):

bWaitForBuffer – Indicates to the Playback component that it should way until buffering of further multimedia objects is complete

This state information provides a mechanism with which the reception and playback of multimedia objects can be synchronized.
The multimedia object contains information required to properly configure the audio and video decoders, and this information is passed to the respective decoder.

After configuring the audio and/or video decoding components of the multimedia object player, the object player may choose to either delay playback until the multimedia object buffers in memory have filled or may begin playback immediately while requesting the next multimedia

Google Exhibit No. 1008
Page 023

object concurrently. This decision can be based on the speed at which the multimedia objects are retrieved versus the playback time of each multimedia object, the latency of requests for multimedia objects or the number of multimedia objects that can be stored in memory at once.

Following the reception and parsing of the first multimedia, its audio And/or video content are hence parsed and played back. In the audio and/o video content of the first and every subsequent multimedia object, the approach taken is to decode sufficient audio frames that their total duration is as long as the display time of the associated video frame and processing time of the next audio frame. By interleaving the processing between several audio frames and a single video frame, the decoder can perform both audio and video decoding in a single thread.

The retrieval and playback of multimedia objects continues until the last multimedia object in the sequence has been completely retrieved and its playback has finished.

The state information described also provides a mechanism which can be used to skip backwards and forwards through the multimedia object sequence. By changing the state information and restarting the retrieval of multimedia objects, the playback of the objects can be

Google Exhibit No. 1008
Page 024

repositioned to any multimedia object in the sequence.

In FIG. 5, a large scale "live content" application is illustrated. A central server indexing host 17 manages all of the available content and the content servers 7 through which the content is made available. Remote transcoding and multimedia object creating servers 18 that provide continuously updated content must register this content with the indexing host 18. The transcoding servers 18 must also keep the central indexing server 17 updated with the latest multimedia object sequence indices, to allow distributed wireless clients 8 to begin playback of any live content with minimal delay.

The URLs of any content servers supporting a particular broadcast would be pre-registered in a table on the indexing server 17.

Content servers 7 accept and store live content being transmitted from transcoding servers 18. They can also store non-live archive multimedia content, but in a live content type application, they need only cache the most current window of multimedia objects.

Content servers 7 are distributed in such a fashion that allows wireless clients 8 a fast and low latency host connection. Content servers 7 could all be connected in a LAN, but for large scale operations, they could have any distribution on the wired Internet. The wireless

Google Exhibit No. 1008
Page 025

client 8 can receive the content directly 19 from a content server 7 or indirectly 20 through a cellular network proxy server 9.

The central indexing host 17 accepts requests from clients 8 for multimedia content 21. The indexing host 17 must reply with the most suitable content server 7 for the client 8. This can either be done in a round-robin fashion or other factors can be included such as the location of the client 8 relative to available content servers 7. Other information such as the server load and network congestion of each content server 7 can be taken into account.

The central indexing host 17 also authenticates 22 clients 8 as they request available content and specific pieces of content. The authentication process is designed in such a way that the content servers 7 do not need to maintain a list of authorized clients 8 and the content available to them. Instead the indexing host 17 authenticates the client 8 and provides the client 8 with an encrypted string that is eventually decrypted by the content server 7. This string is the encrypted form of the catenation of the content name or description, and the current UTC date-time and an interval of time for which the client 8 is authorized to access the multimedia content. The string is designed to allow the client 8 to access and playback multimedia objects received from a designated content server 7.

Google Exhibit No. 1008
Page 026

The indexing host 17 may also provide the client 8 with other information about the multimedia content, along with the encryption string, such as a description of the source, copyrights, and subtitle data-sources.

FIG. 6 illustrates MPEG4 video decoding process as outlined by the MPEG-4 Committee for recovering video object planes (VOPs) data from the coded bit stream. These steps of video decoding are followed in general by the video decoding process of multimedia players 10 and 11. The decoding process is composed of three major sections: shape, motion, and texture decoding.

Coded Bit Streams: The video stream is parsed and demultiplexed to obtain shape, motion, and texture bit streams. Each stream has a decoding process needed in order to reconstruct the VOPs.

Shape Decoding: Binary shape decoding is based on a block-based representation. The primary coding methods are block-based context-based binary arithmetic decoding and block-based motion compensation.

Variable Length Decoding: Shape information, motion vectors, and the quantized DCT coefficients are encoded using variable length codes.

Google Exhibit No. 1008
Page 027

Differential DC coefficients in intra macroblocks are encoded as variable length codes. The final DC value is the sum of the differential DC value and the predicted value. The AC coefficients and non-intra block DC coefficients use a different variable length code.

Inverse Scan: Coefficients are scanned during the encoding process for two reason - to allocate more bits to high energy DCT coefficients during quantization and to turn the two dimensional array (8x8) into a one dimensional array. The reverse process (i.e. inverse scan) is used on the decoding size to ensure proper dequantization and to restore the two dimensional information. There are three types of scans used; Alternate-Horizontal scan, Alternate-Vertical scan, and the Zigzag scan. The type of scan used during the decoding process will depend on the type of coefficients being decoded.

Inverse AC and DC Prediction: The prediction process is only carried out for intra macro blocks. Previous intra macro blocks are used for forward prediction in order produce subsequent macro blocks. This optimization process is used to predict both DC and AC coefficients.

Inverse Quantization: The two-dimensional array of coefficients produced by the inverse scan is inverse quantized to produce the reconstructed DCT coefficients. The process is trivial; it is basically a multiplication by the quantizer step size. A variable quantizer step

Google Exhibit No. 1008
Page 028

size can be produced by using a weighted matrix or a scale factor in order to variably allocate bits during the encoding/decoding process.

Inverse DCT (IDCT): An inverse DCT is applied in order to recover the VOP from the frequency domain (i.e. DCT coefficients) into the spatial domain (i.e. pixel values). Note that in the texture decoding process, the luminance and chrominance components of the VOP (i.e. Y, Cb, Cr components) are quantized at different rates in order to reach a higher compression rate (which is the powerful aspect of the DCT transform when used in compression).

Motion Decoding and Compensation: Motion compensation is another technique used to achieve high compression. The algorithm used by MPEG-4 is block-based motion compensation to reduce the temporal redundancy between VOPs. Motion compensation in this case is two fold: if is used to predict current VOP from previous VOP, and to interpolate prediction from past and future VOPs in order to predict bi-directional VOPs. Motion vectors must be decoded to predict movement of shapes and macroblocks from one VOP to the next. Motion vectors are defined for 8x8 or 16x16 regions of a VOP.

As exemplified by MPEG4 Simple Profile decoding diagram FIG. 6, but common to other video codecs, two types of Video Object Planes

Google Exhibit No. 1008
Page 029

(VOP) are handled in digital video decoding. Video streams must begin with a frame that makes no temporal reference to any earlier frames or an Intra-Frame (I-Frame). A second type of VOP that allows temporal reference to the previous frame in the stream are known as Predicted Frames (P-Frames). Macroblocks within P-Frames may contain motion vectors to enable motion correction from the previous frame. These macroblocks often contain pixel residue information which includes corrections to the predicted pixels.

Motion compensation must occur for many of the macroblocks within P-Frames and is a critical component of any video decoding mechanism. Motion vectors can be compressed using Huffman codes. These are binary Variable Length Codes (VLC) which represent values occurring with high probability with shorter binary length than values which occur with less probability. The rapid decoding of VLCs is critical to any decoding application on constrained devices. The video decoding process operating on the multimedia object players 10 and 11, decodes these VLCs in a novel use of Huffman codebooks.

Theoretical Huffman codebook process reads bits from the packet bitstream until the accumulated bits match a codeword in the codebook. This process can be thought of as logically walking the Huffman decode tree by reading one bit at a time from the bitstream,

Google Exhibit No. 1008
Page 030

and using the bit as a decision Boolean to take the 0 branch (left side) or the 1 branch (right side). Walking this binary tree finishes when the decoding process hits a leaf in the decision tree - the result is the entry number corresponding to that leaf. Reading past the end of a packet propagates the 'end-of-stream' condition to a decoder.

The novel approach taken to decode VLCs by the video decoding process operating on the multimedia object players 10 and 11, is illustrated in FIG. 7, and can be precisely described as follows:

> Bits are read off the stream into an integer buffer (N). The number of bits read is equivalent to the length of the longest code in the VLC codebook. The roof of logarithm (base 2) of N is taken. Based on the result, N is shifted and used as an index into an array containing the true value indicated in the codebook and the true length of the code. The number of bits indicated as the true length is then removed from the video stream and processing continues. An example is provided:
>
> > Table B-7 of the MPEG4 Standard (Conf. [1]) contains the Code/Value(s) pair:
> >
> > Code: 0000 0101 Values MBType: 2 CBPC: 0b11

The maximum length of a code in Table B-7 is 9. The above code would be read off the bit stream as (N:=) 0000 0101X (where X is a

Google Exhibit No. 1008
Page 031

'Do Not Care' bit). The roof of logarithm (base 2) of N is found to be T he value 4 is then used to identify the array in which N is used as an index to locate the appropriate decoded value. N can also be shifted to remove irrelevant bits allowing the lookup array to be smaller.

This novel approach provides a very low time complexity and due to the nature of Huffman codes, a great majority of codes can be decoded with the first few tables providing a greater cache hit ratio.

Following the reading and processing of motion vectors off the video stream, motion correction must take place. Due to the high latency of memory often used in constrained devices, random memory access and non-contiguous memory access must be minimized. FIG. 8 is a flow diagram, describing video decoding process in the multimedia object players 10 and 11, which illustrates that a texture buffer large enough to contain 4 luminance and 2 chrominance blocks (the dimensions of a macroblock exemplified in the MPEG4 specification) is used to store the predicted pixels from a reference frame. This texture buffer is much smaller then the original video frame and decreases the amount of reading from and writing to non-consecutive bytes within the reference and output video frames. All pixel residues are applied to the texture buffer which is then copied to the output frame. This

Google Exhibit No. 1008
Page 032

method of processing P-frames is optimal in situations where the main processing unit has sufficient cache to store the texture information of the entire Macroblock. In cases where the limited device has very little or no on-die cache, it may be preferable to avoid using a macroblock texture buffer. Also, macroblocks with motion vector information contain pixel residue values that are often distributed in a much smaller range of values than the pixels of a texture. In cases where the device is unable to decode the video stream in real-time, a faster but less accurate IDCT algorithm can be used to process these residue values. Furthermore, to minimize the effect of the less accurate IDCT algorithm, this step is taken first on chrominance pixel residues, but can also occur for luminance pixel residues as required.

The motion vector information associated with a macroblock often references a point between pixels on the reference VOP. This requires that decoders perform bilinear interpolation between pixels. This is a time consuming process requiring the sampling of four source pixels, four additions and a single divide operation for every output pixel. In addition to various arithmetic optimizations performed the video decoding process of the multimedia object players 10 and 11, shown in the flow diagram of FIG. 9, uses faster motion compensation without bilinear interpolation when less quality but faster processing is

Google Exhibit No. 1008
Page 033

required.

Digital video codecs define Luminance and Chrominance values within a given subrange of values, MPEG4 uses [0, 255]. This allows the decoding software to store the Luminance and Chrominace pixels within a single byte of data with the correct precision. However, during the decoding process, values outside the [0,255] are often generated during motion compensation and in the inverse DCT steps. Attempting to store values outside this range results in a single byte overflows causing graphical errors in the final video output. Clipping these values and modifications to the dequantization process can be very time consuming and can result in decrease of output correctness.

FIG. 10 is a flow diagram illustrating novel optimization for the dequantization step of digital video decoding in the multimedia object players 10 and 11. The novel optimization requires the a reduction in pixel accuracy but allows values outside the range [0, 255] to be represented in a byte field without an overflow. Through analysis of various video samples, it has been found that the range [-128, 383] is sufficient to store nearly all potential resulting Luminance and Chrominance pixel values. By taking the original pixel value, adding 128 to it and dividing the result by two, values in the [-128, 383] may be represented in the [0, 255] with a decrease in accuracy of 50%.

Google Exhibit No. 1008
Page 034

This decrease in luminance and chrominance accuracy is not a factor on many limited devices as the RGB color resolution is often in the 4-bit to 18-bit range. As an example an input pixel (nInputPixel) in the range [-128, 383] is converted into the alternate format for storing in a byte field (nbOutputPixel): byte nbOutputPixel = (nInputPixel + 128) / 2.

One of the most processing intensive steps of the decoding process occurs during the IDCT step. The use of an effective integer-based algorithm is an absolute requirement when the decoding occurs on constrained devices. The Chen's IDCT algorithm is optimized, but the processing time consumed by the standard Chen implementation is to great for real-time decoding on limited devices. Hence, FIG. 11 is a flow diagram illustrating a novel use of Chen's algorithm in the multimedia object players 10 and 11. Here several different simplified versions of Chen's algorithm be implemented, based on the energy or distribution of input DC and AC coefficients. This can result in reduced video output quality, but the effect is mitigated by giving a higher-quality preference to luminance blocks. Reduced color definition is often not as noticeable on constrained devices, and allows the chrominance blocks to be decoded with less precision. The IDCT process can be further optimized by recording which rows of the input matrix to the IDCT are populated with values. This same mechanism

Google Exhibit No. 1008
Page 035

can be used to ignore certain input values of insufficient energy to make a very noticeable impact on the output image and further decrease processing time.

In a limited device, the memory access required in the YUV to RGB conversion process can be sufficiently long to consume more time then any other step in the video decoding process. The video decoding process in the multimedia object players 10 and 11 uses a further step of scaling to reduce this processing, as the display size is often not the exact size of the video output. The YUV to RGB conversion and scaling steps can be combined into a single step to decrease memory access and increase the speed of video output. Several YUV to RGB functions are available providing decoding times of varying speeds and quality as well as scaling ranges. FIG. 12 is a flow diagram showing video decoding of the YUV to RGB step in the in the multimedia object players 10 and 11, as follows:

1) Separate YUV to RGB and scaling functions for cases where scaling up is required and where scaling down is required. Distinct optimizations are available for each method and added speed can be attained by separating the functionality between several different functions.

Google Exhibit No. 1008
Page 036

2) When scaling up is required, a minimum amount of reading from the source Luminance and Chrominance planes is desired. This is accomplished by iterating through pixels in the source plane. A fixed number of Luminance and Chrominance values in a column are read and the resulting RGB values computed for each pixel position. The pixel values are then copied in a looping fashion first by column, then by row to the output plane. This provides a way to read a single input value which may result in many output values in the output plane when scaling up.

3) Similarly, when scaling down is required a minimum amount of reading from the source Luminance and Chrominance planes is desired. This is accomplished by iterating through pixel positions in the output plane and calculating the source pixel in the input plane. This provides a way to read a single input value for every output value and minimizes the number of input-plain reads that are necessary.

4) The YUV to RGB conversion step is such a time consuming one that methods of improving the speed of computation at the expense of output quality have been implemented. Improvements in speed can be obtained by sampling only a subset of the chrominance pixels, avoiding pixel clipping or calculating the Red and Blue values for only a subset of output pixels. All of these methods are used together to

Google Exhibit No. 1008
Page 037

provide several quality levels in the YUV to RGB step.

Hence, FIG. 13 is a flow diagram summarizing the short-cut optimization processing by the video decoding process used in the multimedia objects players 10 and 11. State information is maintained about the quality levels with which the current video stream is processed. On very limited devices short-cuts in the decoding process must be made to allow the device to maintain synchronicity between the audio and video playback. These short-cuts are specifically designed to drastically reduce the number of computations necessary at certain critical steps in the video decoding process at the cost video output quality. This mechanism allows video decoding to scale in complexity based on the processing power of the device being used.

It has been found that three quality levels tests at each critical step appear to yield the best results. The highest quality is consistent with the video codec specification and displays a correct image. A medium quality level indicates that certain time consuming short-cuts are made with some impact on image quality. A low quality level indicates that drastic reductions in display quality are made to improve processing time - the output video can be unrecognizable at times and as a result this level is used only in drastic cases of a sudden drop in processor availability.

Google Exhibit No. 1008
Page 038

A final option exists is to avoid the processing and displaying of some or all P-Frames. This is only an option in video streams where I-Frames occur at regular intervals. Given the wide variety of processing capabilities in limited devices, this implementation strongly suggests the creation of multimedia objects from video streams with transcoder 2 specifying very regular I-Frames so that devices of very limited processing power are able to provide the client 8 with occasional frame changes.

The state information is composed of a series of integers correspond to various steps in the decoding process and define the quality at which the decoder should perform several steps. The implemented system in the multimedia players 10 and 11 consists of six of these integers:

- nYUVtoRGBQuality - Quality of the YUV to RGB conversion process
- nLumaIDCTQuality - Quality of the Inverse DCT function for Luminance blocks
- nChromaIDCTQuality - Quality of the Inverse DCT function for Chrominance blocks
- nLumaMCQuality - Quality of motion compensation for Luminance blocks
- nChromaMCQuality - Quality of motion compensation for Chrominance blocks

Google Exhibit No. 1008
Page 039

- nFrameRateQuality - Defines the allowance to drop frames (from a single P-Frame occuring before an I-Frame up to dropping all P-Frames)

In addition to the set of integers defining the actual quality at various steps, a single integer representing the current quality level of the overall decoding is used (named nVideoQuality in this instance). Each step quality has a very limited number of possibilities (HIGH, MEDIUM, LOW, etc), however, nVideoQuality can take on many values. At each value of nVideoQuality, a ruleset defines quality of each of the above step qualities. At the highest value of nVideoQuality, all step qualities are set to maximum. As an nVideoQuality is decreased, the step qualities are incrementally reduced according to the ruleset.

Some states of quality levels are less preferable to others. For example, it is not preferable to render many frames at the lowest quality setting of nLuminaIDCTQuality – it is instead more preferable to drop frames if there is insufficient processing capability to perform nLumaIDCTQuality at a higher quality. The ruleset is designed to take these possibilities into account.

FIG. 14 illustrates the general steps of audio decoding followed by the audio decoding process of the multimedia object players 10 and 11.

Google Exhibit No. 1008
Page 040

The first step in AAC audio decoding (bit-stream de-multiplexing), which is common to other digital codecs, is to establish frame alignment. This involves finding the AAC sync word and confirming that the AAC frame does not contain any errors, if error checking is enabled in the frame. Once the frame sync is found, the bitstream is de-multiplexed or unpacked. This includes unpacking of the Huffman decoded and quantized scale factors, the M/S synthesis side information, the intensity stereo side information, the TNS coefficients, the filter bank side information and the gain control words.

Next the quantized spectral coefficients are Huffman decoded. Each coefficient must be inverse quantized by a 4/3 power nonlinearity and then scaled by the quantizer step size.

The Huffman codebooks used to decode digital audio in the multimedia object players 10 and 11 of FIG. 1, are very different from those used for digital video, but they are very similar to Huffman codebooks used in other digital audio codecs. A novel simplification of variable length fields (VLCs), used in audio decoding by the multimedia object players 10 and 11, is illustrated in FIG. 15, that allows the decoding of a single VLC value with a single array lookup. The novel approach taken is as follows:

Bits are read off the stream into an integer N. The number of bits

Google Exhibit No. 1008
Page 041

read is equivalent to the maximum number of bits in the longest codeword in the codebook. The first binary 0 is then located starting from the highest bit. The left-based index of this first 0 is then used to remove out all the previous 1s and N is shifted and used as an array Index.

For example, the AAC standard's 2nd Codebook contains the Code/Value pair:

Code: 11110110, Value: 77

The maximum length of a code in the 2nd table is 9 so when read from the BitStream the above code would appear as:

11110110X (Where X is a "Do Not Care" bit)

The ZeroPosition of the above integer is found to be 4. The ZeroPosition is then used to mask off the 1 bits previous to it yielding the integer "0101X". This can then be used as an index to an array or be shifted to remove the irrelevant bits allowing the lookup array to be smaller.

The next standard audio decoding step, M/S synthesis, conditionally dematrixes two channels into a stereo pair. The samples my already represent the left and right signals, in which case no computation is necessary. Otherwise the pair must be de-matrixed via one add and one subtract per sample pair in order to retrieve the proper channel

Google Exhibit No. 1008
Page 042

coefficients.

Intensity stereo identifies regions in a channel pair that are similar, except for their position. Left-channel intensity regions must have inverse quantization and scaling applied. Right-channel intensity stereo regions use the left-channel inverse quantized and scaled coefficients, which must be re-scaled by the intensity position factors. Hence the net complexity of intensity stereo is a savings of one inverse quantization per intensity stereo coded coefficient. The next standard step, temporal noise shaping (TNS), has a variable load, depending on the number of spectral coefficients that are filtered.

Finally, the Inverse Modified Discrete Cosine Transform (IMDCT) transforms the spectral coefficients into time-domain samples. For fixed-point implementations it is required that any round-off noise is less than 1/2 LSB after the transform result is rounded to linear 16-bit values. Fixed-point realizations using 24 bit words are sufficient.

FIG. 16 illustrates Intermediate 23 and Final 24 optimizations for the digital audio IMDCT step used by the audio decoding process in the multimedia object players 10 and 11. The audio decoder of the multimedia object players 10 and 11, combines the use of a specific Inverse Fast-Fourier Transform with Pre- and Post- processing steps.

Google Exhibit No. 1008
Page 043

This method produces a simplified IMDCT algorithm with O(n * log(n)) runtime. This method can also incorporate the use of various IFFT algorithms based on the sparseness of input.

The following the steps describe the implementation:

[0] The IMDCT algorithm accepts an input array X of spectral coefficients in the frequency domain and outputs an array of amplitude values in the time domain twice the size of the input. The implementation of the AAC Low Complexity codec requires that the IMDCT algorithm accept input array lengths of 128 or 1024 Real values and results in an output of 256 or 2048 Real values. In the following steps, N refers to the size of the output (256 or 2048), Im(X) returns the imaginary component of some variable X and Re(X) returns the real component.

[1] The (N/2) input spectral coefficients are converted into complex numbers and stored into an array C of size (N/4). There are many approaches to this step, however, the approach taken in the described implementation pairs coefficients with one coefficient becoming the real component and one becoming the imaginary component of a complex number. The following pseudo code describes this step:

```
for (n=0; n<N/4; n++) {
    Re(C[n]) = X[N/2 - 2*n];
```


Google Exhibit No. 1008
Page 044

```
    Im(C[n]) = X[n];
}
```

[2] This result is then multiplied with scaled complex numbers on the unit circle yielding an array of size N/4. This step is described with the following pseudo code:

```
for (n=0; n<N/4; n++) {
    Re(Z) = SQRT(2/N) * cos(2*Pi*(n + 1/8) / N)
    Im(Z) = SQRT(2/N) * sin(2*Pi*(n + 1/8) / N)
    Re(C[n]) = Re(C[n]) * Re(Z)  -  Im(C[n]) * Im(Z);
    Im(C[n]) = Re(C[n]) * Im(Z)  +  Im(C[n]) * Re(Z);
}
```

[3] The resulting array of complex numbers is then passed into an Inverse Fast Fourier (IFFT) Algorithm. A fixed-point IFFT algorithm is used to allow processing of IMDCT on devices which lack floating point capabilities. Most mobile devices do not allow floating point computations, and of those that do allow floating point, it is usually too slow.

Due to the properties of the inverse Fourier transformation, the transformation can be calculated in a fixed point manner. In a fixed point transformation, the input is be scaled by multiplying the input values by a scale factor and then the correct output is found by

Google Exhibit No. 1008
Page 045

multiplying by the reciprocal of the scale factor. Therefore a scaling operation is applied before and after the IFFT. A scale factor which is a power of two is chosen so that the scaling and re-scaling operations can be accomplished by bit shift operations. Bit shifts are among the fastest operations for CPUs.

[4] Following the Inverse FFT step, elements from the complex array C must again be multiplied by complex numbers as in step [2].

[5] The values from the resulting complex array C are then stored into an array of Real numbers x of size N. The following pseudo code demonstrates the process:

```
for (k=l; l<N/8; l+=2) {

        x[2*l  ] = Im(C[N/8+l]);

        x[2*l+1] = Re(-C[N/8-1-l]);
        x[2*l+2] = Im(C[N/8+1+l]);

        x[2*l+3] = Re(-C[N/8-2-l]);

        x[2*l+N/4  ] = Re(C[l]);

        x[2*l+N/4+1] = Im(-C[N/4-1-l]);

        x[2*l+N/4+2] = Re(C[l+1]);

        x[2*l+N/4+3] = Im(-C[N/4-2-l]);

        x[2*l+N/2  ] = Re(C[N/8+l]);

        x[2*l+N/2+1] = Im(-C[N/8-1-l]);

        x[2*l+N/2+2] = Re(C[N/8+1+l]);

        x[2*l+N/2+3] = Im(-C[N/8-2-l]);
```


Google Exhibit No. 1008
Page 046

```
    x[2*I+N/2+N/4  ] = Im(-C[I]);
    x[2*I+N/2+N/4+1] = Re(C[N/4-1-I]);
    x[2*I+N/2+N/4+2] = Im(-C[I+1]);
    x[2*I+N/2+N/4+3] = Re(C[N/4-2-I]);
}
```

As can be seen in FIG. 16, several steps in the IMDCT process can be combined. The goal of combining steps is to reduce the number of memory accesses needed to decode a frame of audio. The flow on the right shows the steps as they occur in the decoder.

In summary then, the novel optimization of the IMDCT step in audio decoding shown by FIG. 16 pertains to combining steps on the Final 24 optimization side:

1.Re-order, pre-scale and twiddle: The method loops over the input data, and each datum is complex-multiplied by the twiddle factor, and is then re-scaled by doing a bit shift operation. However, the twiddle factor is already bit-shifted so it can be treated as a fixed-point number, so the scaling operation's bit shift is partially performed by the twiddle factor itself. The relevant twiddle factors are stored in an array table. Once the complex multiplication and scaling are done, the

Google Exhibit No. 1008
Page 047

resulting values are stored in the re-ordered location in the IFFT input array.

2. Perform the fixed-point integer inverse Fourier transform. This transformation is the same as the transformation in the pre-combined flow.

3. Re-scale, re-order, post-twiddle, window and overlap: Combining these four operations into one step replaces four array accesses with one, and some of the multiplications are also combined into single bit shifts. This method loops over the IFFT output array, and performs four operations in each iteration of the loop: the post-twiddle and rescale are combined, because the post-twiddle uses a twiddle factor table which is already bit-shifted. Windowing is combined in this step also, with window values coming from either a table or a fast integer sine calculator. Finally, values are overlapped and stored in the correct location in the output array.

FIG. 17 illustrates simplified input shortcut processes that are specific to AAC Low Complexity (LC) profile which are used in the audio decoding process of multimedia players 10 and 11. Note, that the Mid/Side, Intensity and Temporal Noise Shaping steps, marked with cross hatches above, are optional. In cases where these three features are not present, audio decoding can further combine other steps in a novel way. These steps are marked in grey in FIG. 17. If

Google Exhibit No. 1008
Page 048

these other steps are combined, there are no dependencies within a frame until we reach the IFFT step within IMDCT itself. Therefore operations between noiseless decoding and the pre-IFFT operations within IMDCT itself are combined, minimizing memory access.

IMDCT has four different window shapes which are common in other digital audio codecs: long only, long start, long stop, and eight short. Of these four window sequences, only one (long only) has non-zero data in the entire output synthesis window. In the case of AAC, however, the output synthesis window always has 2048 output values.

Window shape	*Non-zero byte range*
LONG ONLY	0 – 2047
LONG START	0 – 1599
LONG STOP	448 – 2047
EIGHT SHORT	448 – 1600

For some window shapes, the calculations can be short-cut, avoiding the post-twiddle, windowing, re-ordering, scaling and overlapping steps entirely.

IMDCT permits two different window types: Kaiser-Bessel Derived (KBD) windows and Sine windows. KBD uses a complicated formula which cannot be computed in real-time, and is always used as a table. Sine windows are also used from tables in most implementations.

Google Exhibit No. 1008
Page 049

However, on a mobile device, which generally has a very small on-CPU memory cache, frequent accesses to a sine window value table will cause cache misses and degraded performance.

As an alternative to using a sine lookup table to compute windowing, the FIG 18 shows the audio decoder of the multimedia objects players 10 and 11, using a bit-operation based Taylor computation, as follows:

1. Use trigonometric identities to express the sine calculation in terms of a sine in the range of 0 to π / 2. Call the resulting angle X.
2. Calculate X*X. Call this value S.
3. Calculate the result as X * (256 - S * (43 - (S << 1))
4. The result produces a window value in the range of 0 to 255, allowing fast windowing without the use of lookup tables.

5. The bit shift operations in Step 3 can be further combined with other fixed-point multiplication steps.

FIG. 19 is illustrates further IMDCT short window processing for even greater efficiency by the audio decoding process of multimedia players 10 and 11. In a sequence of eight short windows, the input of 1024 values is divided into eight short windows of 128 values, and IMDCT, windowing and overlapping is performed on each of these short

Google Exhibit No. 1008
Page 050

windows. Each window of 128 values results in a synthesis output window of 256 values. These are then overlapped, resulting in non-zero values in the range of 448 to 1600.

The approach taken is to do every one of the IMDCT operations in sequence, rather than in parallel, storing the IMDCT results directly into the regions of the output array which will be zeroed. The output values are then windowed and overlapped. After all the eight short windows are completed, the regions of the synthesis output window which are always zero can be disregarded, due to the window shape shortcut method described above.

Finally, FIG. 20 illustrates an interleaved detection process in the audio decoding of received multimedia objects 25 by the multimedia object players 10 and 11. FIG. 20 illustrates the placement of gaps 26 at detected frames of low total energy 27 are they are detected during audio decoding by the multimedia object players 10 and 11. Hence, playback is then controlled so that the gap will occur during that frame, which may be dropped, so that synchronization with video is not lost.

In FIG. 1 the multimedia object player 10 is a downloadable Java (J2M2) applet and the described audio and video decoder optimizations and strategies, FIG. 7-13 and FIG. 15-20, as applied to

Google Exhibit No. 1008
Page 051

standard MPEG4 and AAC decoding make it possible for the multimedia object player 10 to playback live music and video, at acceptable frame rates (5-15fps), on limited, cell phone handsets. Java cannot take advantage of hardware capabilities in processing the huge number of calculations and variables required for either MPEG4 decoding or AAC decoding. Hence, the required optimizations for multimedia player 10, to permit the playback of AAC and MPEG4 on current mobile client hardware, are a source of technological innovation and advance.

The foregoing is intended, along with the drawings, to illustrate the preferred embodiment of the invention. Those skilled in the art will be able to devise numerous arrangements which, although not explicitly shown or described herein, embody the principles of the invention and are within their spirit and scope as defined by the following claims.

Google Exhibit No. 1008
Page 052

What is claimed is:

1. A method of transmitting multimedia to wireless clients, wherein the multimedia transmission method depends on: the creation of multimedia objects from existing multimedia files or dynamically from live multimedia streams; a direct request and transmission of just the multimedia objects created from existing multimedia files or dynamically created multimedia objects from live multimedia streams by wireless client-based, multimedia object players; and, a continuous playback of the received multimedia objects by wireless client-based multimedia players that are specifically designed to play continuous sequences of the multimedia objects.

2. The method 1, running on a distributed network system for multimedia-on-demand, utilizing a centralized content server, an indexing host, multimedia object creator and transcoder for live broadcast applications or to transcode and create multimedia objects from archived multimedia files, and distributed content servers involving high capacity cellular network proxy servers and mobile clients running downloaded java applets or embedded or downloaded non-java multimedia object players.

3. The method 1, wherein the transmission of said multimedia

Google Exhibit No. 1008
Page 053

objects is by protocols such as:

HTTP, FTP, IMAP4 and NNTP, which have the capability to serve files in a directory structure;

and where HTTP 1.1 is used and allows pipelined connections over persistent TCP connections, multimedia object players can request many multimedia objects in rapid succession.

4. The method of claim 1, wherein the wireless client-based multimedia players are downloadable java applet multimedia object players or non-java multimedia object players, or embedded multimedia object players.

5. A method of creating multimedia objects, where, in the case of a live multimedia stream, the input multimedia stream is first transcoded into a optimal audiovisual format such as MPEG4/AAC and at an optimal encoding rate reflecting available cellular network bandwidth, then dynamically converted into multimedia objects by splitting the encoded stream into specified intervals, and then immediately transmitted to wireless clients to distributed content servers transmitting the recently created multimedia objects to wireless clients; alternatively, in the case of converting an archived multimedia file, the input multimedia stream is first transcoded into a optimal audiovisual format such as MPEG4/AAC and at an optimal

Google Exhibit No. 1008
Page 054

encoding rate reflecting available cellular network bandwidth, and then converted into multimedia objects by splitting the encoded stream into specified intervals.

6. The method of 5, wherein the dynamically created multimedia objects are maintained by content servers serving the multimedia objects to wireless clients, as a windows of multimedia objects, during transmission to wireless clients.

7. The method 5, wherein the input multimedia stream is scanned after specified intervals for the next I-frame, and the multimedia segment is split at that next I-frame to create another multimedia object.

8. The method 5, wherein the input multimedia stream can be in analog audiovisual format or a variety of digital audiovisual formats, including MPEG4, MPEG1, MPEG2, MOV, AVI, WMV, ASF, and higher encoded MPEG4, or just audio formats, including analog audio, mp3, AMR, Windows Media Audio, RealAudio and higher encoded AAC.

9. The method 6, wherein a window of multimedia objects for live transmission is created and comprised of a small series of multimedia objects, which can be incremented and decremented as newly created objects are introduced to the window or transmitted to wireless clients.

Google Exhibit No. 1008
Page 055

10. The method 5, wherein multimedia objects are identified when they are created by the multimedia object creator with an Internet address that includes such information as:
the transport protocol;
the varying host URL, if there are many content servers involved as in a live broadcast application, of the transmission server or content server directly serving the wireless client;
the name of the multimedia object sequence or broadcast;
the number of multimedia objects in the sequence;
and, the multimedia object's sequence number.

11. The method 5, whereby multimedia objects are spit from multiple MPEG4 composite layer streams by scanning time intervals and splitting them at next I-frames.

12. The method 5, whereby audio media objects are split from a single audio stream by splitting at set time intervals.

13. A method of wireless client side processing of multimedia objects by multimedia object player, wherein :
the Identification of the multimedia object is parsed and, the total number of multimedia objects within the Identification path is determined, or the number of multimedia objects in window is

Google Exhibit No. 1008
Page 056

determined for live applications;

heap memory allocations for said multimedia objects and meta-data are determined;

to create a buffer on the wireless client for more than one multimedia object;

to identify multimedia object playing, multimedia object receiving and multimedia wait for states for the multimedia object sequence;

to hence use these states as a mechanism to synchronize the reception and playback of multimedia objects;

to pass this information onto the audio and/or video decoding components of the multimedia player to properly configure them to uniquely process the sequence of multimedia objects.

14. The method 13, whereby, following configuration of audio and/or video decoding components for a specific sequence of multimedia objects, the multimedia object player can delay playback until the multimedia object buffers in the wireless client memory have filled or can begin playback immediately while requesting the next multimedia object;

and, whereby, the multimedia object player decision can be based on the speed at which the multimedia objects are retrieved versus the playback time of each multimedia object, the latency of

Google Exhibit No. 1008
Page 057

requests for multimedia objects, or the number of multimedia objects that can be stored in wireless client memory at once.

15. The method 13, following the parsing of the first multimedia object, its audio and video contents of the first and each subsequent multimedia object in the sequence are decoded and played back, whereby sufficient audio frames are decoded that their total display time is as long as the associated video frame and processing time of the next audio frame; and, whereby interleaving the processing between several audio frames and a single video frame, the multimedia object player can perform audio and video decoding in a single thread.

16. The method 13, whereby state information, also provides a mechanism that can be used to skip backwards and forwards through a multimedia object sequence, wherein changing the state information and restarting retrieval of multimedia objects, repositions playback from any multimedia object in the sequence; and , wherein the transmission is a live transmission, state information can reposition playback from any multimedia object within a current window.

17. A method for processing the large scale distribution of multimedia content in the distributed network being managed by an indexing host server, wherein: the indexing host registers all URLs of content servers

Google Exhibit No. 1008
Page 058

supporting particular live multimedia object transmissions and archived sequences of multimedia objects;

remote transcoding/multimedia object creating servers provide registered updateds of multimedia object sequence indices to the indexing host;

remote transcoding/multimedia object creating servers also register the sequence indices of the most recent windows of live content multimedia objects with the indexing host; wherein content servers accept and store the most current window of live content multimedia objects or the most recent non-live archives of multimedia object sequences;

content servers transmit their multimedia directly to wireless clients, or indirectly through cellular network proxy servers;

and whereby, the indexing host verifies the wireless client;

the indexing host accepts requests from wireless clients for multimedia content;

the indexing host determines the most suitable content server for the wireless client;

and, the indexing host provides the wireless client with a decryption string for the requested multimedia content.

18. A method of optimized video decoding in decoding Variable Length Codes (VLCs) in Huffman codebooks which are used to

Google Exhibit No. 1008
Page 059

compress Motion Vectors for motion compensation occurring in many macroblocks within P-frames, whereby, bits are read off the main video stream into an integer buffer (N);

the number of bits read is equivalent to the longest code in the VLC codebook;

the roof of logarithm (base 2) of N is taken;

based on the result, N is shifted and used as an index into an array containing the true value indicated in the codebook and the true length of the code;

the number of bits indicated as the true length is then removed from the video stream and processing continues;

said optimized video decoding using a texture buffer large enough to 4 luminance and 2 chrominance blocks (the dimensions of a macroblock exemplified in MPEG4 specification) to store predicted pixels from a reference frame;

said texture buffer decreases the amount of reading from and writing to non-consecutive bytes within the reference and output video frames;

all pixel residues are applied to the texture buffer which is then copied to the output frame; to use a faster but less accurate IDCT algorithm with the process if the wireless handset cannot decode the

Google Exhibit No. 1008
Page 060

video stream in real-time, to process these residue values;
furthermore, to minimize the effect of the less accurate IDCT algorithm but using this process first on the chrominance pixel residues;

said optimized video decoding processing faster motion compensation without bilinear interpolation when less quality but faster processing is required;

said optimized digital video decoding performing optimizations in pixel processing and dequantization, whereby original luminance and chrominance values are taken and 128 added and the result divided by 2;
values in the [-128,383] range are then represented in the [0,255] range, decreasing luminance and chrominance accuracy without significantly affected RGB color resolution in the 4-bit to 18-bit range;

said optimized video decoding processing by optimizing Chen's algorithm, whereby, different simplified versions of Chen's algorithm are used, based on the energy input or distribution of input DC and AC coefficients, whereby, the energy or distribution of DC and AC coefficients is first assessed;
a simplified Chen's algorithm is selected for IDCT processing;

Google Exhibit No. 1008
Page 061

a higher quality preference is given to luminance blocks;
and, the process is further optimized by recording rows of the input matrix to the IDCT that are populated with values.

said optimized video decoding processing in the handling YUV to RGB conversion, whereby, YUV and RGB scaling functions are separated;
when scaling up, pixels are read on the source plane and copied to the output plane;
when scaling down, iteration is performed through pixel positions in the output plane and source pixels are calculated in the input plane; and, sampling is performed on only a subset of chrominance pixels, avoiding pixel clipping or calculating the Red and Blue values for only a subset of output pixels;

said optimized video decoding processing by using short-cuts to permit video decoding to scale in complexity, based on the processing power of the wireless client, whereby, three quality levels are used with high being consistent with a correct image in the digital codec specification;
medium corresponds to some reduction in image quality to reduce processing time;
and low being a drastic reduction in image quality to improve processing time;

Google Exhibit No. 1008
Page 062

wherein a final option is to avoid the processing and display of P-frames when I-frames occur at regular intervals.

said optimized video decoding processing by using short-cuts to permit video decoding to scale in complexity, based on the processing power of the wireless client, where state information defines the quality at which decoding should be performed at several steps of the decoding process;

said state information consisting of six integer value steps defining state: Quality of the YUV to RGB conversion process;
Quality of the Inverse DCT for luminance blocks;
Quality of the Inverse DCT function for chrominance blocks;
Quality of Motion Compensation for luminance blocks;
Quality of Motion Compensation for chrominance blocks;
and, allowance to drop frames (from a single P-frame occurring before an I-Frame up to dropping all P-Frames);

said state information further including a single integer representing the quality level of the overall encoding, wherein, at each value of overall quality, a ruleset defines quality for each of the step qualities;
and, at the highest overall quality, all step qualities are set to maximum;
and, as overall quality is decreased, step qualities are

Google Exhibit No. 1008
Page 063

incrementally reduced according to the ruleset.

19. A method of optimized audio decoding pertaining to a simplification of variable length codes (VLCs) in Huffman codebooks, wherein, bits are read off the audio stream into an integer N;
the number of bits read is equivalent to maximum number of bits in the longest codeword in the codebook;
the first binary 0 is then located starting from the highest bit;
the left-based index of this first 0 is then used to remove all the previous 1s; and, N is shifted and used as an array index;

said optimized audio decoding for optimizations in the IMDCT step, whereby, the Inverse Fast Fourier Transform is combined with pre- and post-processing steps to produce a simplified IMDCT algorithm with O(n*nlog(n)) runtime, which can incorporate various IFFT algorithms based on the sparseness of input, and, which specifically involves the following combination of steps in a final optimization:

a) Re-order, pre-scale and twiddle, whereby, the method loops over the input data, and each datum is complex-multiplied by the twiddle factor, and is then re-scaled by doing a bit shift operation; and, the twiddle factor is already bit-shifted so it can be treated as a fixed-point number, so the scaling operation's bit shift is partially performed by

Google Exhibit No. 1008
Page 064

the twiddle factor itself; and the relevant twiddle factors are stored in an array table; and finally, once the complex multiplication and scaling are done, the resulting values are stored in the re-ordered location in the IFFT input array;

b) Perform the fixed-point integer inverse Fourier transform;

c) Re-scale, re-order, post-twiddle, window and overlap, whereby combining these four operations into one step replaces four array accesses with one, and some multiplications are also combined into single bit shifts; and hence, the method loops over the IFFT output array, and performs four operations in each iteration of the loop: the post-twiddle and rescale are combined; the post-twiddle uses a twiddle factor table which is already bit-shifted;
and, windowing is combined in this step also, with window values coming from either a table or a fast integer sine calculator;
and finally, values are overlapped and stored in the correct location in the output array;

said optimized audio decoding performing simplified input processing when specific to AAC Low Complexity (LC) audio decoding profile, wherein, Mid/Side, Intensity and Temporal Noise Shaping steps, are optional;
in cases where these three features are not present, there are no

Google Exhibit No. 1008
Page 065

dependencies within a frame until the IFFT step within IMDCT itself; and, operations between noiseless decoding and the pre-IFFT operations within IMDCT itself are combined, minimizing memory access.

said optimized audio decoding using an alternative bit-operation based upon Taylor computation, wherein, trigonometric identities are used to express the sine calculation in terms of a sine in the range of 0 to PI/2, resulting in angle X;
X is multiplied by X, resulting in S;
perform a bit-shift operation by calculating X*(256-S*(43-(S<<1));
the result producing a window value in the range of 0 to 255, allowing fast windowing without the use of lookup tables;
and, combining the bit-shift operation with other fixed-point multiplication steps;

said optimized audio decoding using IMDCT short window processing for digital audio decoding, wherein, IMDCT 1024 values are divided into sequences of 8 short windows;

IMDCT window and overlap functions are performed on each short window;
each window of 128 values results in a synthesis output window of 256 values;

Google Exhibit No. 1008
Page 066

these output windows are then overlapped, resulting in a non-zero values in the range of 448 to 1600.

20. A method of low energy gap timing in audio playback, wherein, an interleaved process in audio decoding detects frames of low energy;
audio playback is controlled so a gap will occur during the detected frames, which may be dropped so that synchronization with video is not lost.

Google Exhibit No. 1008
Page 067

Abstract of the Disclosure

A method for multimedia playback and transmission to wireless clients is described. A host webserver transcodes a live digital or analog audio-visual or audio broadcast signal and splits the input stream into small multimedia objects of an efficient compression such as MPEG4/AAC, and then immediately deploys the objects to distributed content servers for a geographically dispersed population of wireless clients. A java applet object player, downloaded to wireless clients at the beginning of the multimedia on-demand session, interprets and decodes the multimedia objects as they are received, using multiple levels of optimization. The applet uses novel video and audio decoding optimizations which can be generically applied to many digital video and audio codecs, and specifically decodes Simple Profile MPEG4 video and Low Complexity AAC audio.

Google Exhibit No. 1008
Page 068

FIG 1

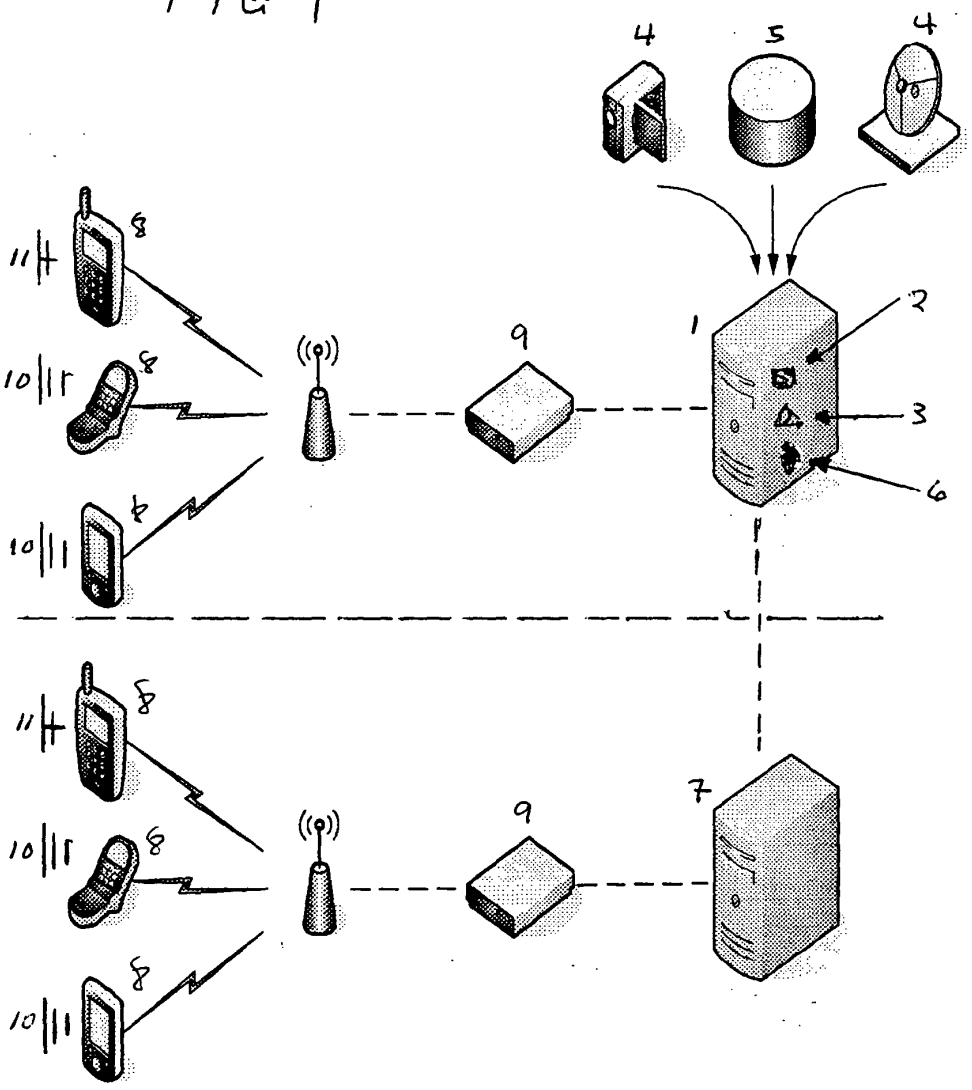


FIG 2

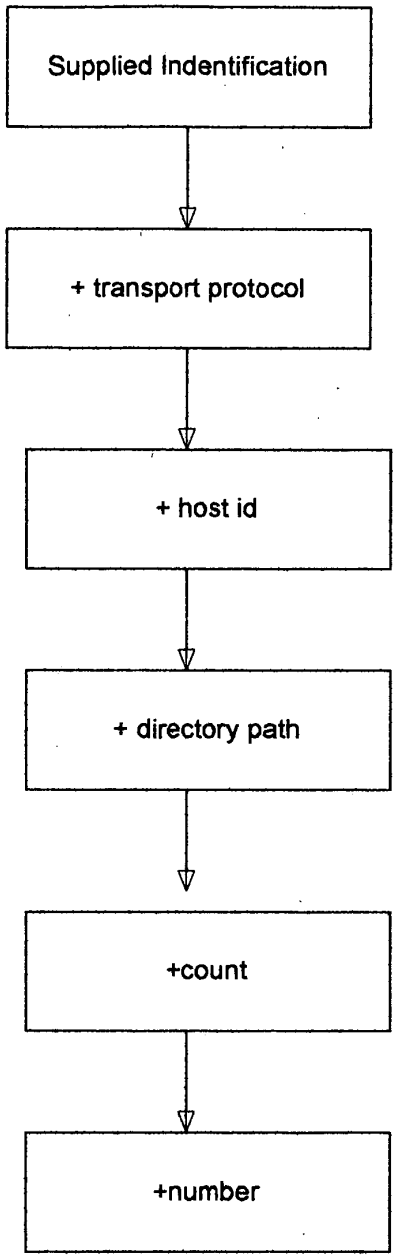

Supplied Indentification
+ transport protocol
+ host id
+ directory path
+count
+number

FIG 3.

MO	Event
0	in window
1	in window
2	in window
3	in window
4	in window
0	deleted
5	in window
1	deleted

FIG 3a

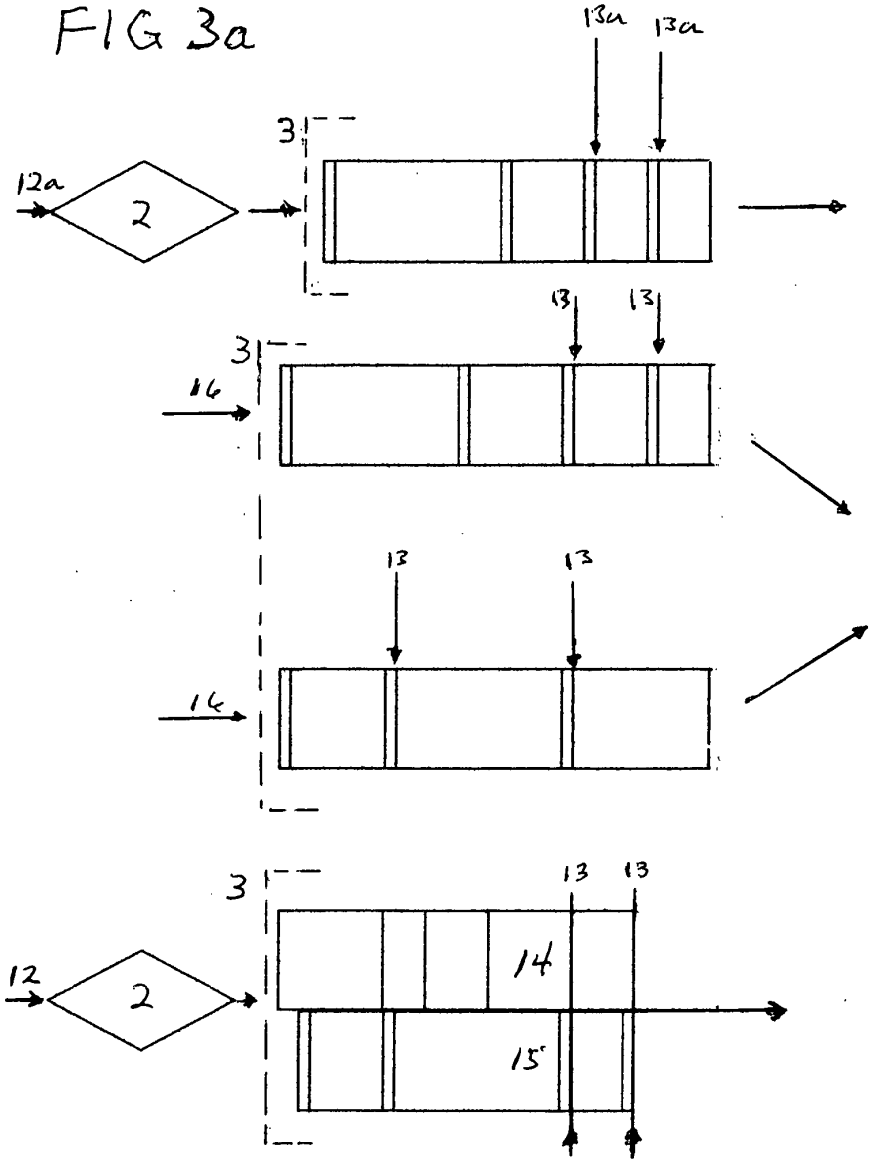


FIG 4

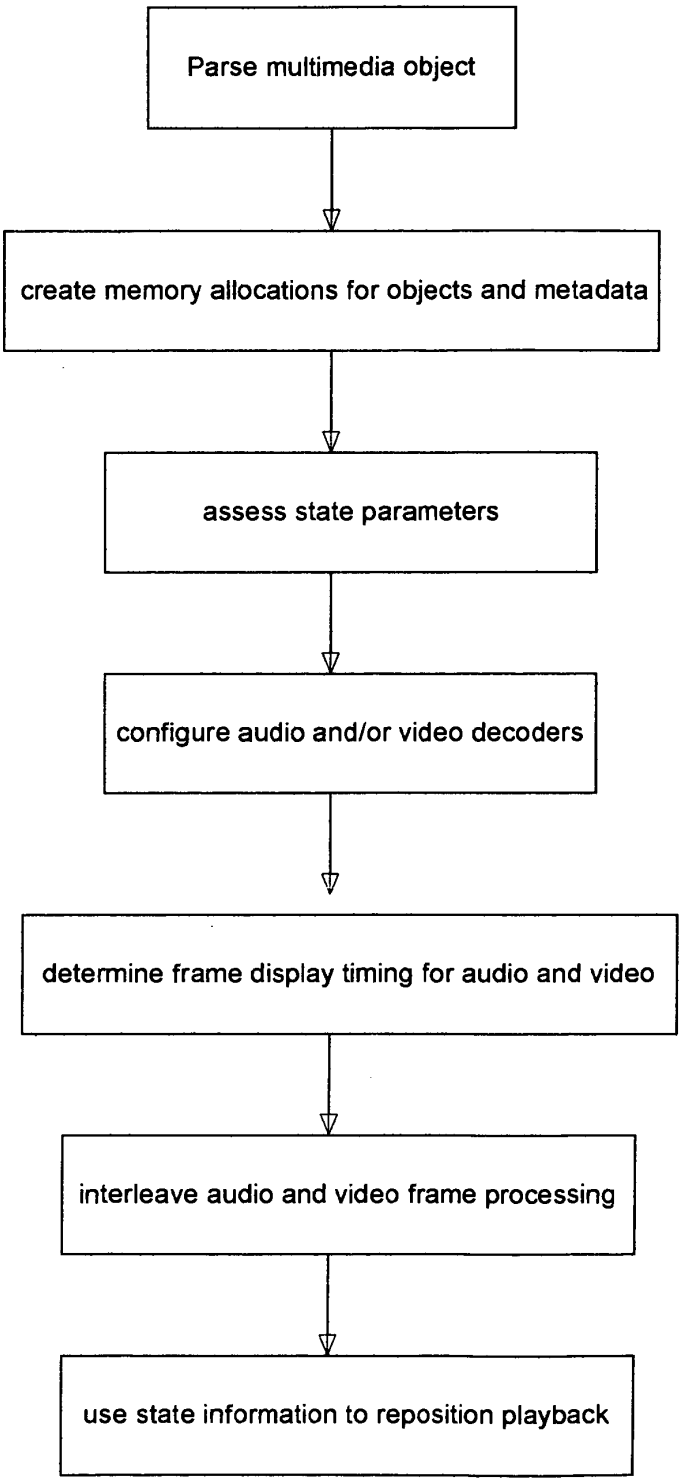

Parse multimedia object
create memory allocations for objects and metadata
assess state parameters
configure audio and/or video decoders
determine frame display timing for audio and video
interleave audio and video frame processing
use state information to reposition playback

FIG 5

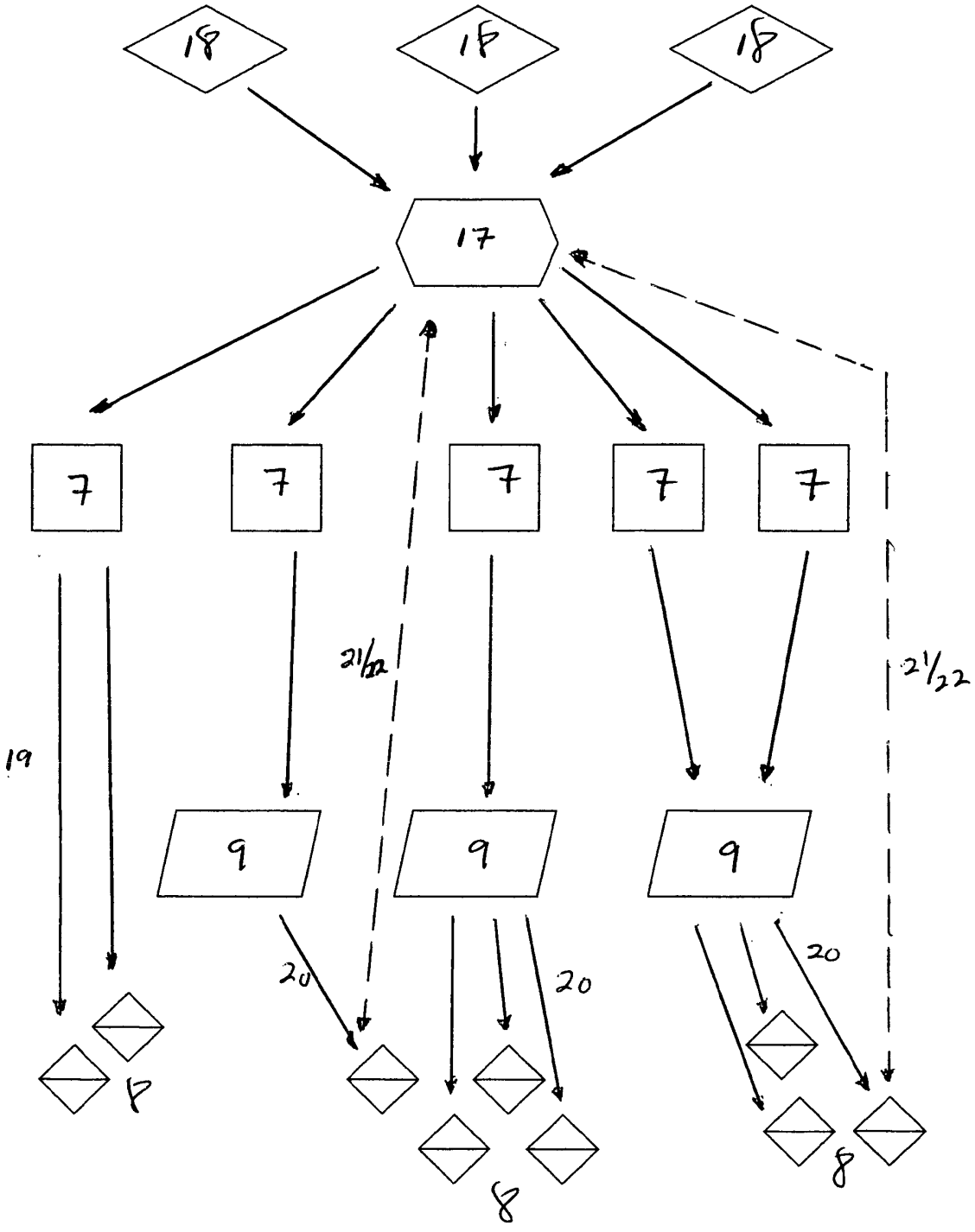


FIG 6

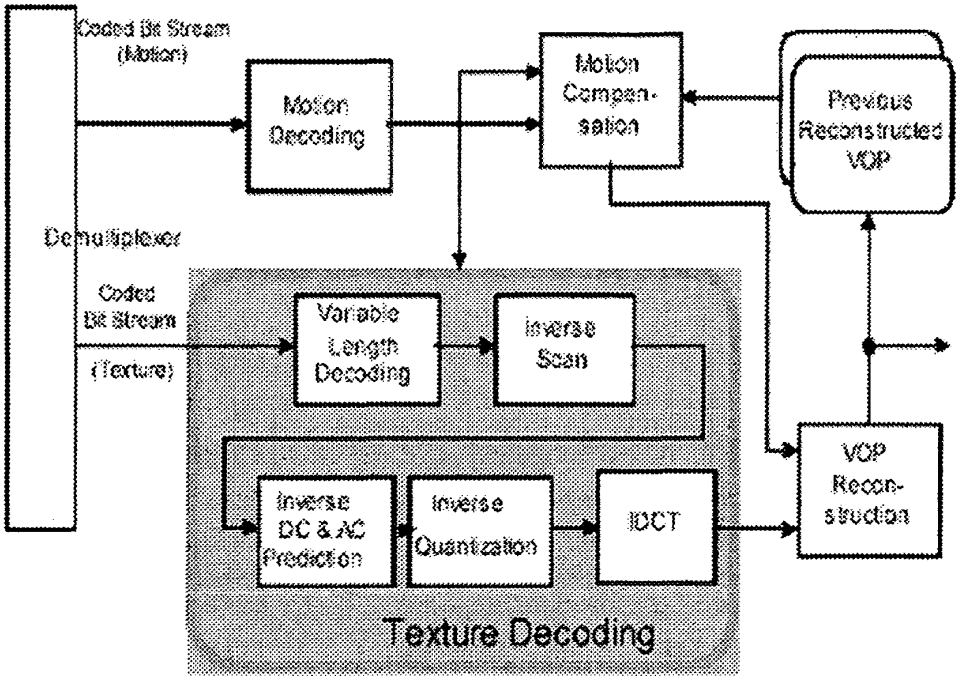

Coded Bit Stream (Motion)
Motion Decoding
Motion Compen-sation
Previous Reconstructed VOP
Demultiplexer
Coded Bit Stream (Texture)
Variable Length Decoding
Inverse Scan
Inverse DC & AC Prediction
Inverse Quantization
IDCT
VOP Recon-struction
Texture Decoding

FIG 7

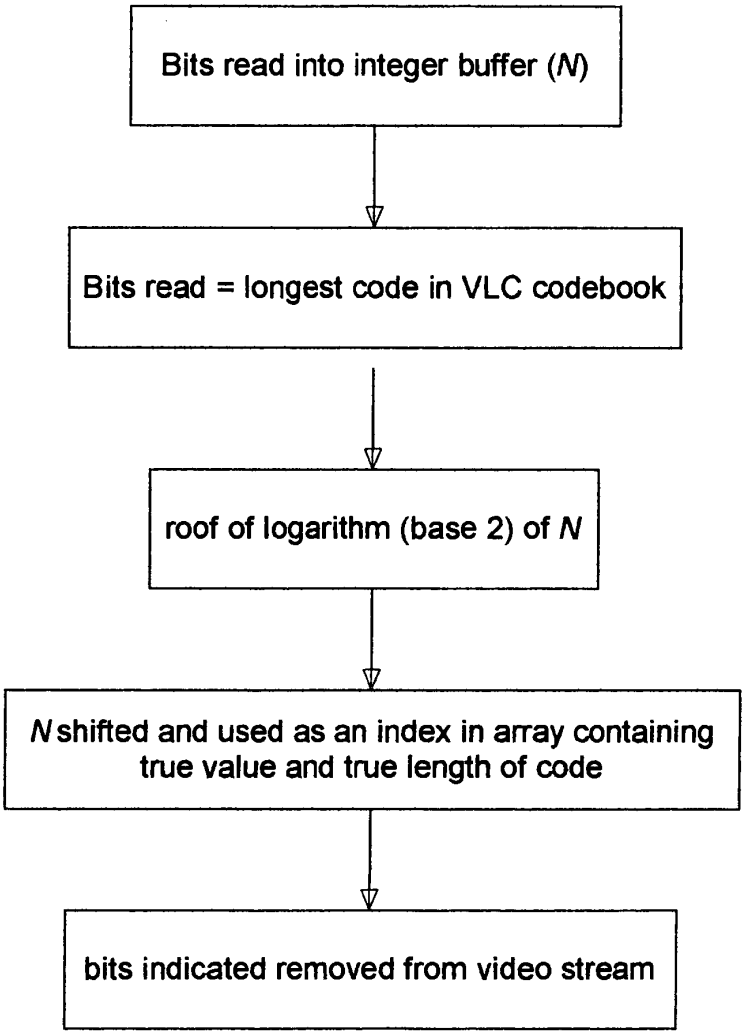

Bits read into integer buffer (N)
Bits read = longest code in VLC codebook
roof of logarithm (base 2) of N
N shifted and used as an index in array containing true value and true length of code
bits indicated removed from video stream

FIG 8

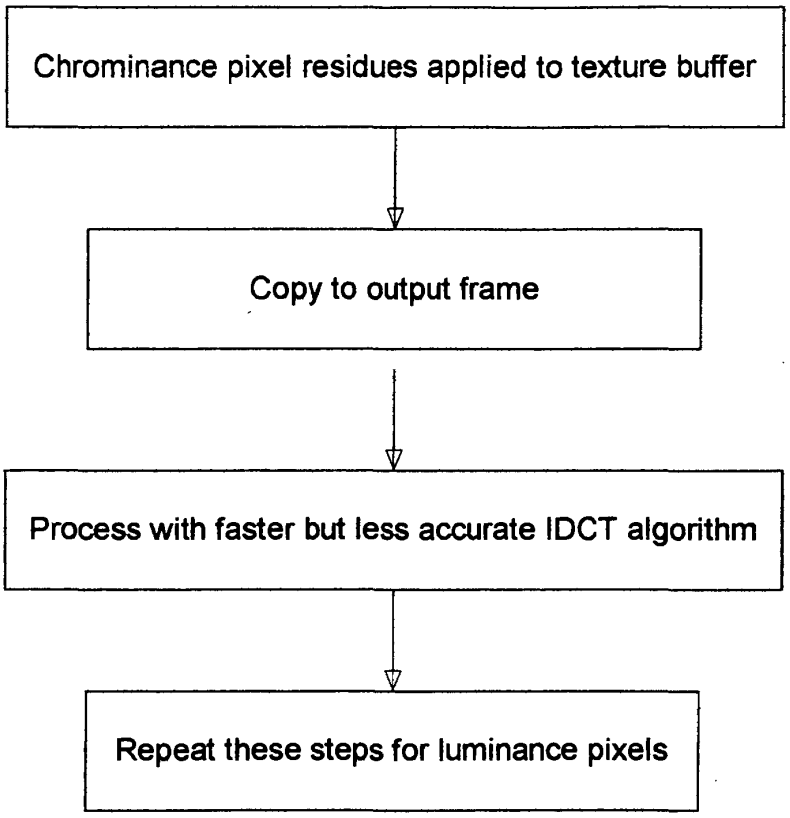

Chrominance pixel residues applied to texture buffer
Copy to output frame
Process with faster but less accurate IDCT algorithm
Repeat these steps for luminance pixels

FIG 9

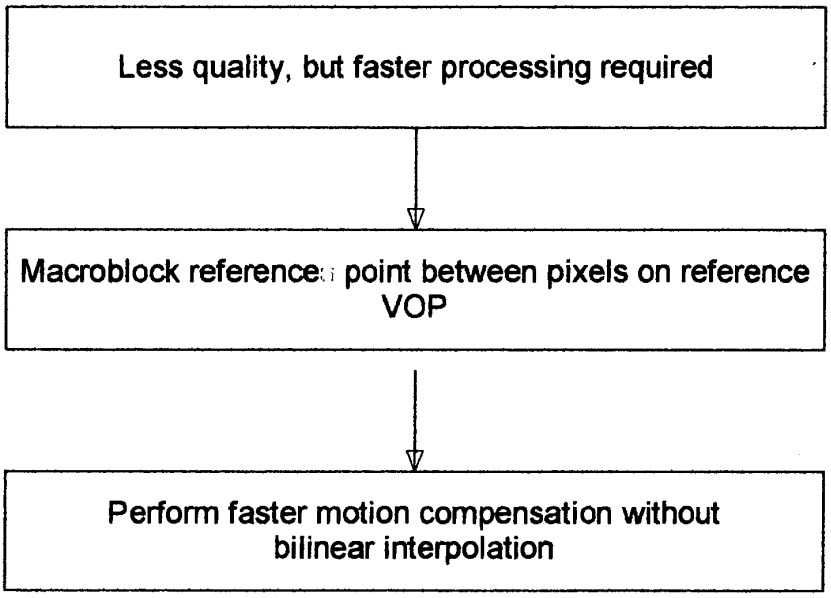

Less quality, but faster processing required
Macroblock reference point between pixels on reference VOP
Perform faster motion compensation without bilinear interpolation

FIG 10

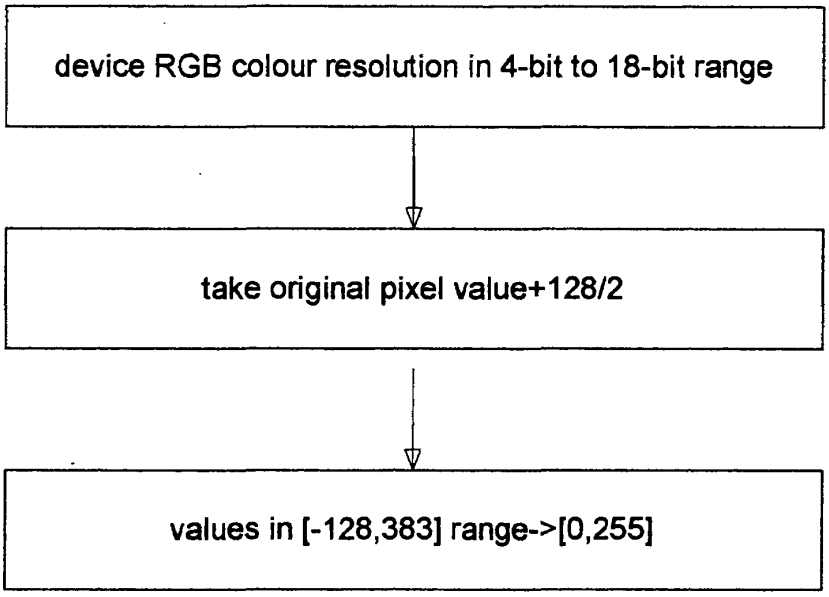

device RGB colour resolution in 4-bit to 18-bit range
take original pixel value+128/2
values in [-128,383] range->[0,255]

FIG 11

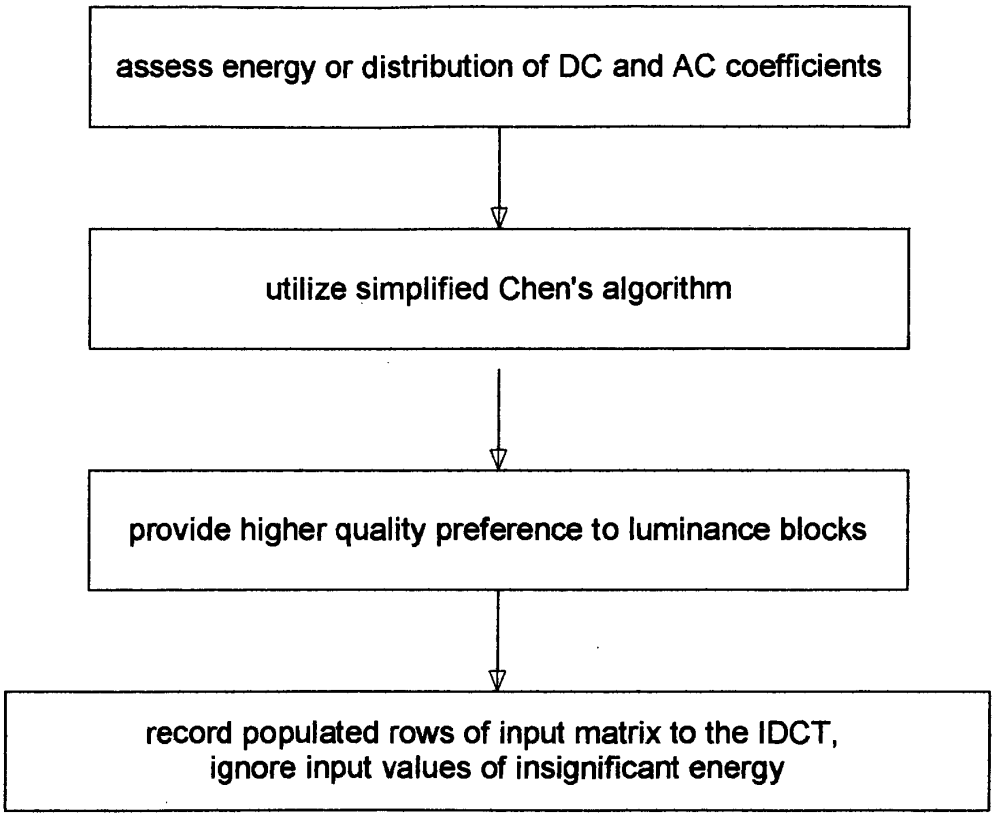

assess energy or distribution of DC and AC coefficients
utilize simplified Chen's algorithm
provide higher quality preference to luminance blocks
record populated rows of input matrix to the IDCT, ignore input values of insignificant energy

FIG 12

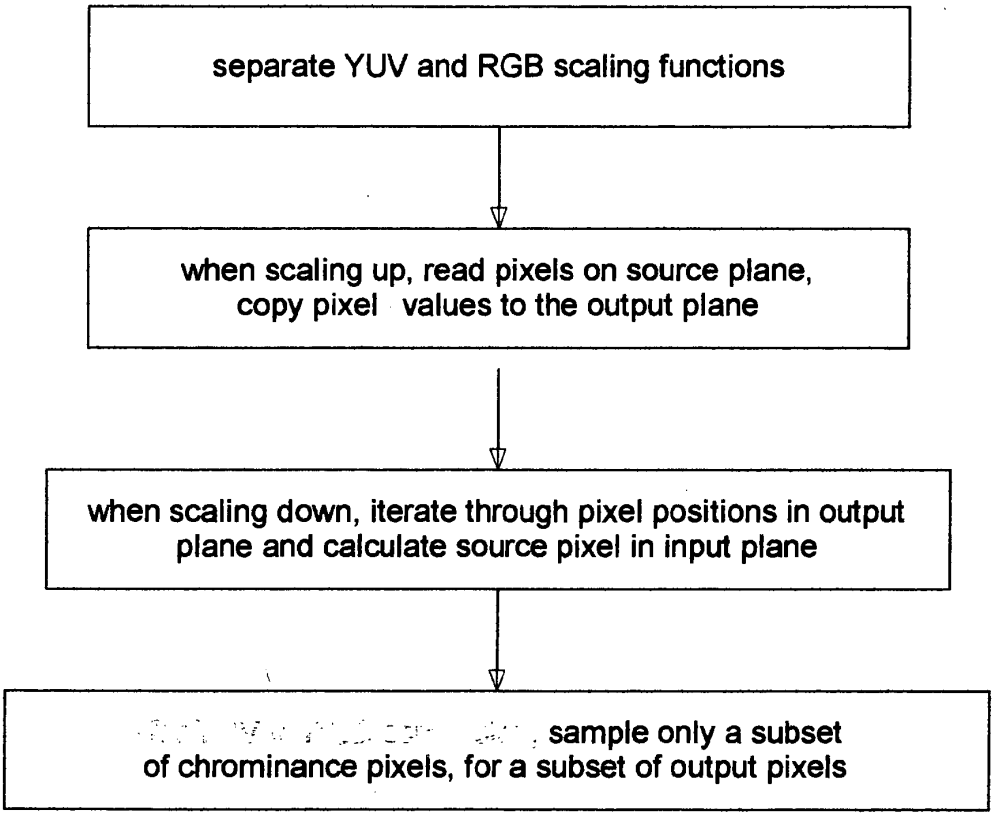

separate YUV and RGB scaling functions
when scaling up, read pixels on source plane, copy pixel values to the output plane
when scaling down, iterate through pixel positions in output plane and calculate source pixel in input plane
sample only a subset of chrominance pixels, for a subset of output pixels

FIG 13

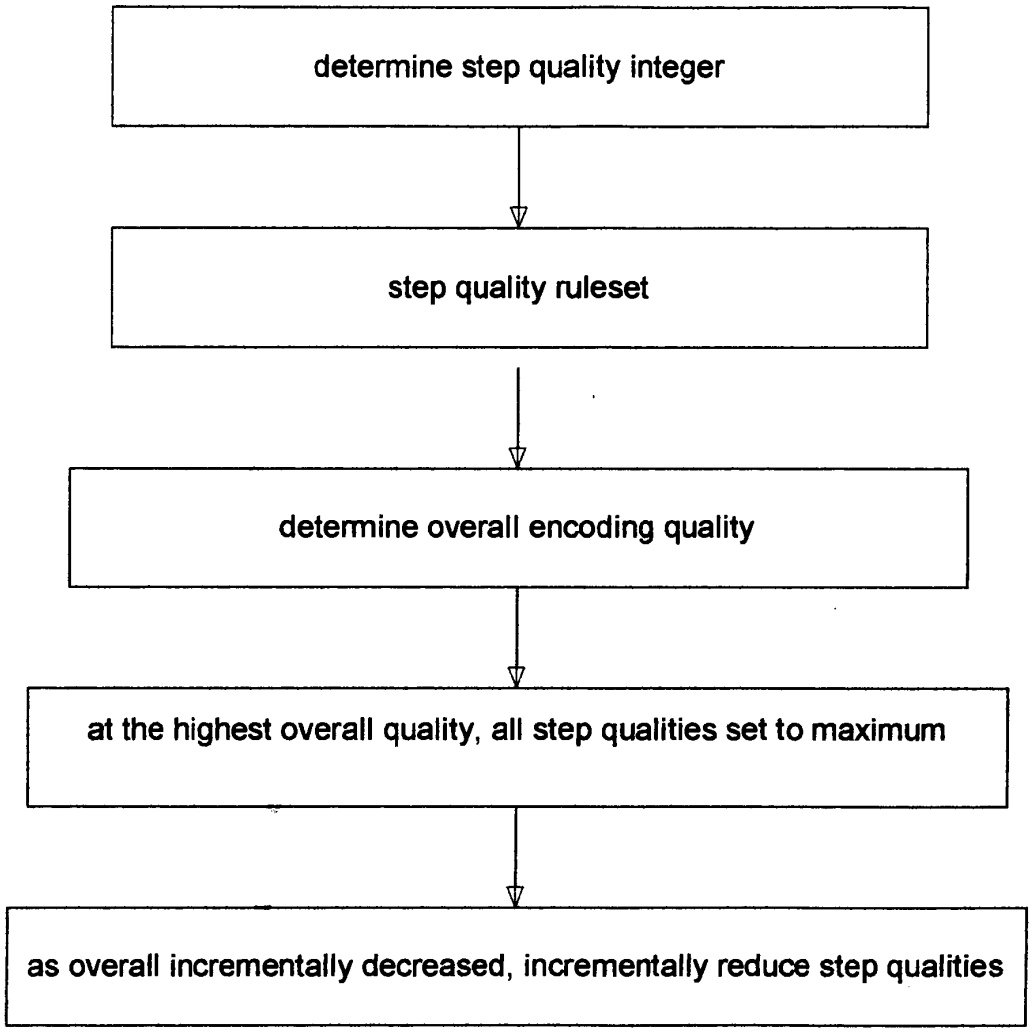

determine step quality integer
step quality ruleset
determine overall encoding quality
at the highest overall quality, all step qualities set to maximum
as overall incrementally decreased, incrementally reduce step qualities

FIG 14

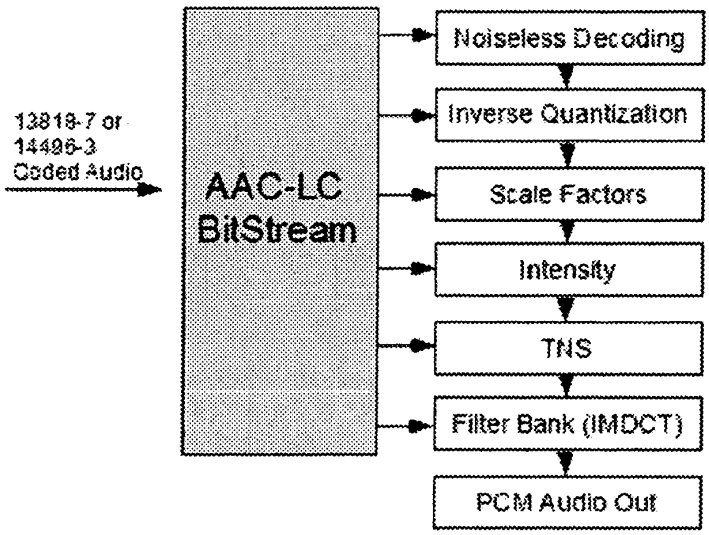

13818-7 or
14496-3
Coded Audio
AAC-LC
BitStream
Noiseless Decoding
Inverse Quantization
Scale Factors
Intensity
TNS
Filter Bank (IMDCT)
PCM Audio Out

FIG 15

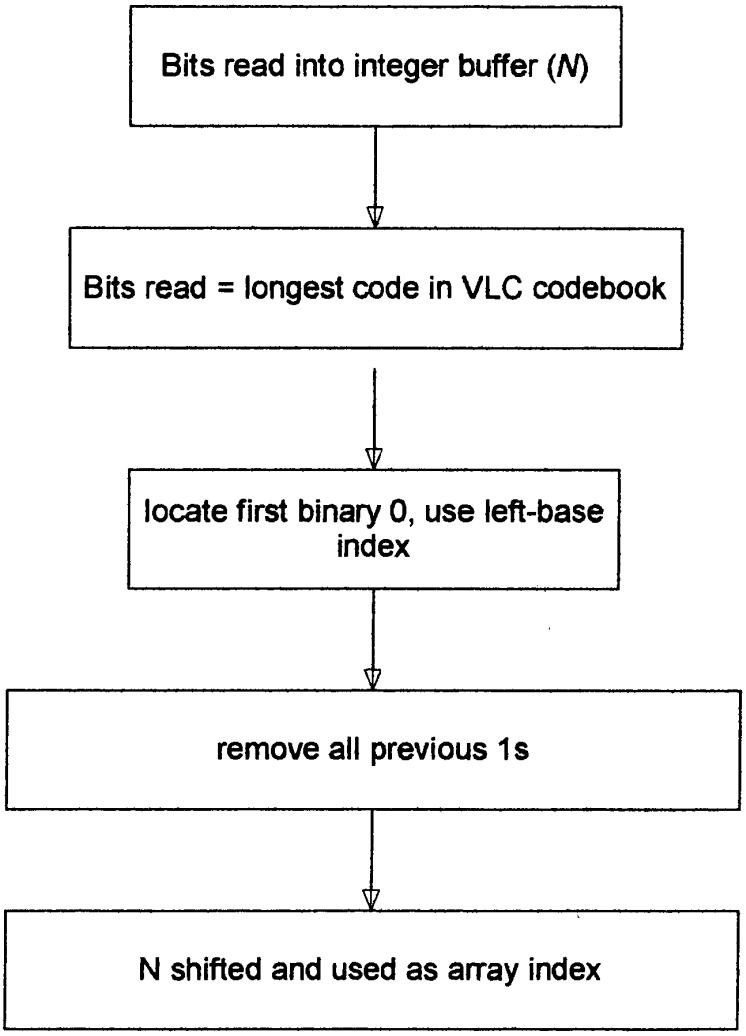

Bits read into integer buffer (N)
Bits read = longest code in VLC codebook
locate first binary 0, use left-base index
remove all previous 1s
N shifted and used as array index

Fig 16

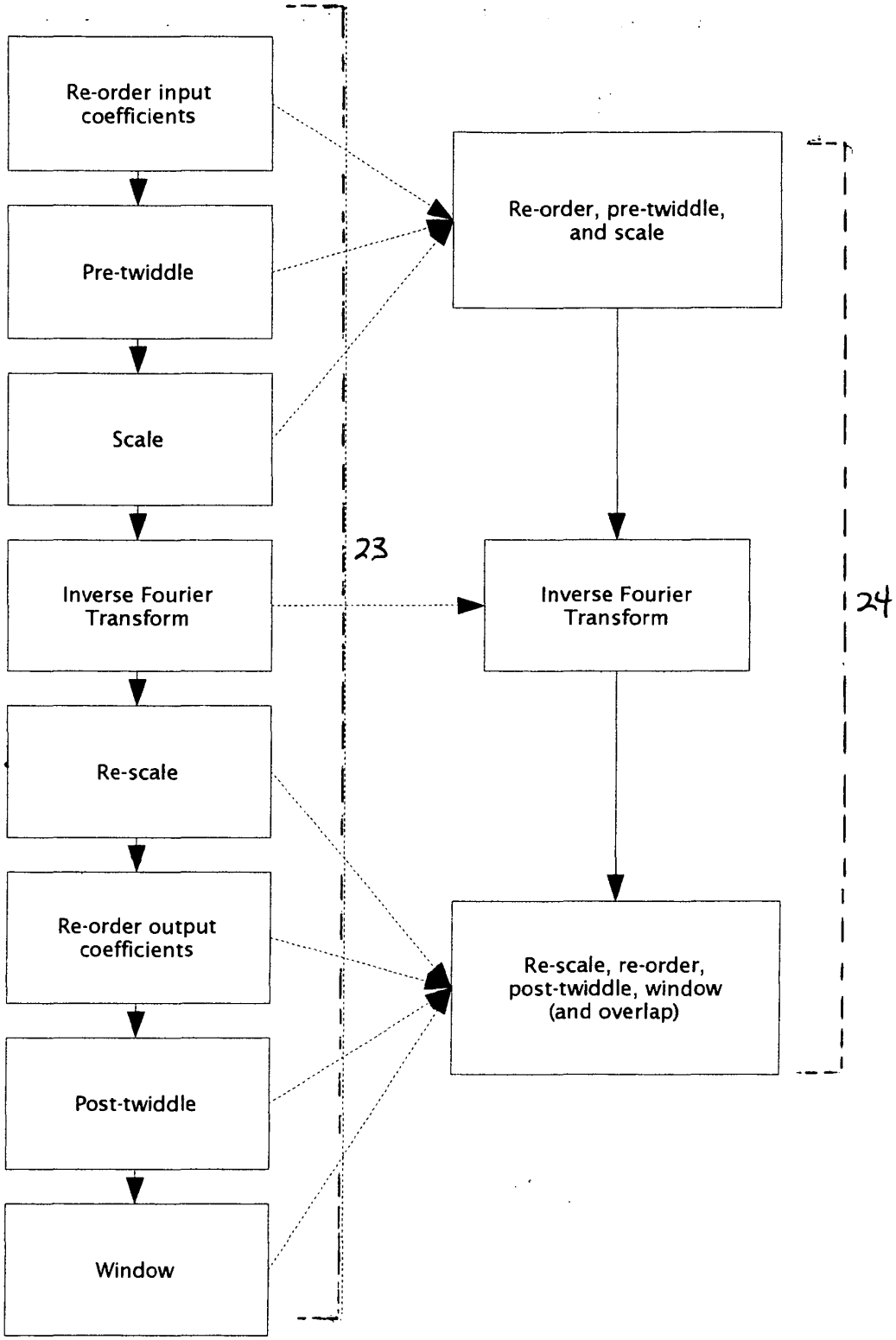

Re-order input coefficients
Pre-twiddle
Scale
Inverse Fourier Transform
Re-scale
Re-order output coefficients
Post-twiddle
Window
23
Re-order, pre-twiddle, and scale
Inverse Fourier Transform
Re-scale, re-order, post-twiddle, window (and overlap)
24

FIG 17

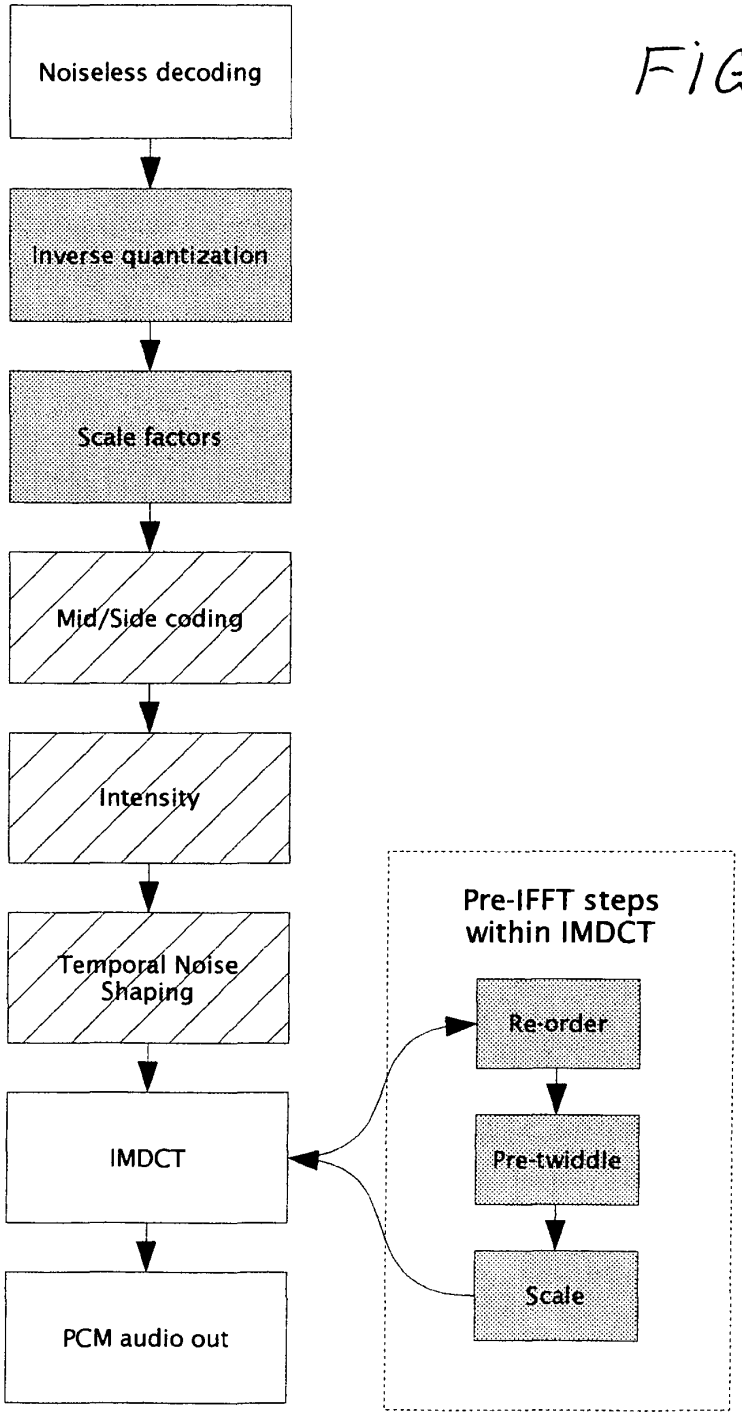

Noiseless decoding
Inverse quantization
Scale factors
Mid/Side coding
Intensity
Temporal Noise Shaping
IMDCT
PCM audio out
Pre-IFFT steps within IMDCT
Re-order
Pre-twiddle
Scale

FIG 18

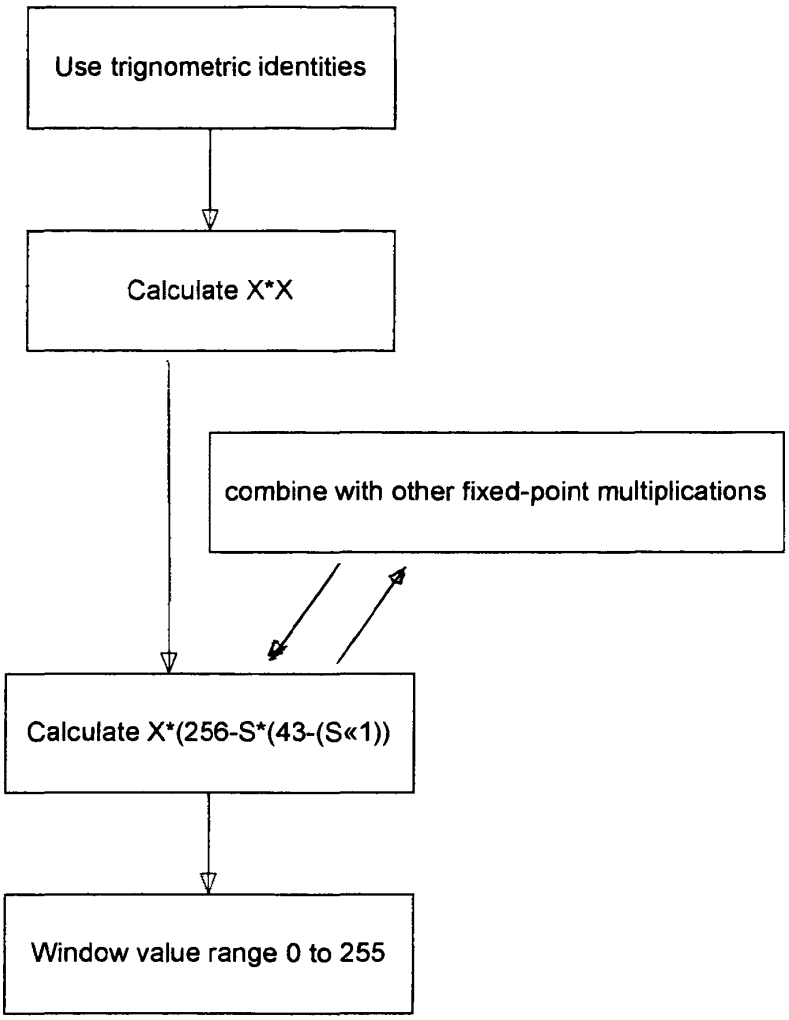

Use trignometric identities
Calculate X*X
combine with other fixed-point multiplications
Calculate X*(256-S*(43-(S«1))
Window value range 0 to 255

Fig 19

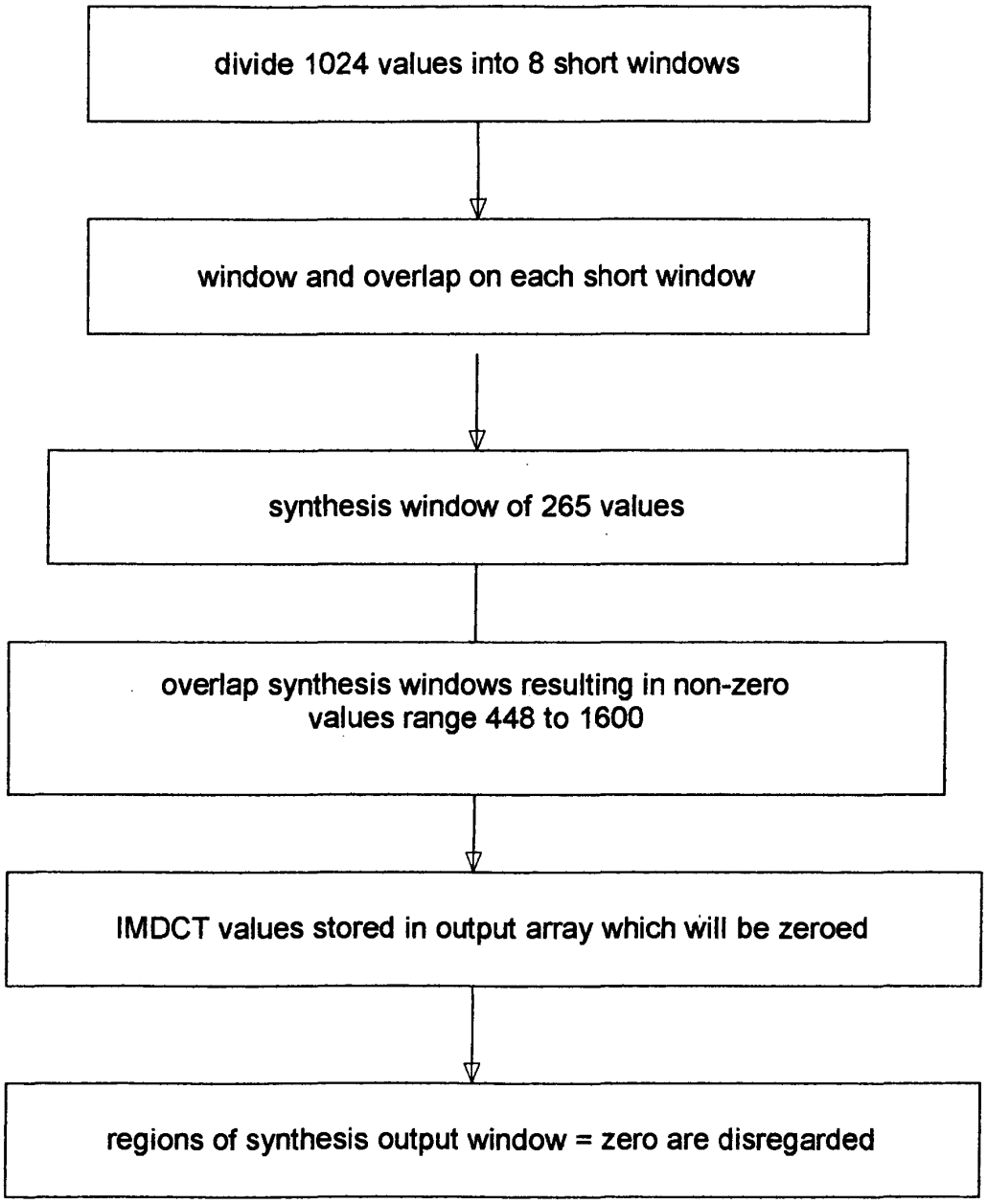
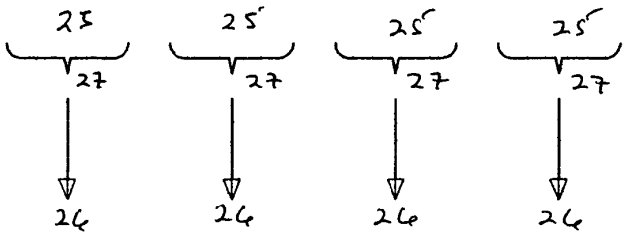

divide 1024 values into 8 short windows
window and overlap on each short window
synthesis window of 265 values
overlap synthesis windows resulting in non-zero values range 448 to 1600
IMDCT values stored in output array which will be zeroed
regions of synthesis output window = zero are disregarded

FIG 20



Electronic Patent Application Fee Transmittal				
Application Number:				
Filing Date:				
Title of Invention:	Multimedia System for Mobile Client Platforms			
First Named Inventor/Applicant Name:	Mark Sinclair KREBS			
Filer:	Yasin Bismilla			
Attorney Docket Number:	2018-00816			
Filed as Micro Entity				
Filing Fees for Utility under 35 USC 111(a)				
Description	**Fee Code**	**Quantity**	**Amount**	**Sub-Total in USD($)**
Basic Filing:				
BASIC FILING FEE-UTILITY	3011	1	75	75
UTILITY SEARCH FEE	3111	1	165	165
UTILITY EXAMINATION FEE	3311	1	190	190
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
		Total in USD ($)		**430**

Electronic Acknowledgement Receipt

EFS ID:	34218295
Application Number:	16181285
International Application Number:	
Confirmation Number:	1045
Title of Invention:	Multimedia System for Mobile Client Platforms
First Named Inventor/Applicant Name:	Mark Sinclair KREBS
Customer Number:	132285
Filer:	Yasin Bismilla
Filer Authorized By:	
Attorney Docket Number:	2018-00816
Receipt Date:	05-NOV-2018
Filing Date:	
Time Stamp:	21:18:42
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Application Data Sheet	2018-00816-ADS.pdf	1818851 e1edda623af10ef6823e32f357c61d3d69de7fe3	no	7

Warnings:

Information:					
2	Oath or Declaration filed	2018-00816-decSigned2.pdf	788198 42eb80e6856fd8775c80c4288cb1281f12cc728a	no	2
Warnings:					
Information:					
3	Certification of Micro Entity (Gross Income Basis)	2018-00816-micros.pdf	445811 405ce4c673b77b9de73152f686680208f38d11a2	no	1
Warnings:					
Information:					
4	Specification	2018-00816-description.pdf	1718700 6c92bb7806cc635c8dab341db94df0d3f8743513	no	51
Warnings:					
Information:					
5		2018-00816-claimsAndAbstract.pdf	475507 3179cd27f50b625e858182c86151bc298122081b	yes	16

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Claims	1	15
	Abstract	16	16

Warnings:					
Information:					
6	Drawings-other than black and white line drawings	2018-00816-drawings.pdf	336262 54ee3dd2da53be77b2c07e5a5b87b0bfae68d39f	no	21
Warnings:					
Information:					
7	Fee Worksheet (SB06)	fee-info.pdf	34932 2050617dbcf42886b4854a78bd7cb986bb5b8367	no	2
Warnings:					
Information:					

Total Files Size (in bytes):	5618261

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 (07-14)
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	2018-00816
	Application Number	
Title of Invention	Multimedia System For Mobile Client Platforms	

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
	Mark	Sinclair	KREBS	

Residence Information (Select One) US Residency ● Non US Residency Active US Military Service

City	Toronto	Country of Residence [i]	CA

Mailing Address of Inventor:

Address 1	169 Browning Avenue		
Address 2			
City	Toronto	State/Province	ON
Postal Code	M4K1W6	Country [i]	CA

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	132285		
Email Address	ip@elan-ip.com	Add Email	Remove Email

Application Information:

Title of the Invention	Multimedia System For Mobile Client Platforms		
Attorney Docket Number	2018-00816	Small Entity Status Claimed	☒
Application Type	Nonprovisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)	21	Suggested Figure for Publication (if any)	1

PTO/AIA/14 (07-14)
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	2018-00816
	Application Number	
Title of Invention	Multimedia System For Mobile Client Platforms	

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	132285		

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 this 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 blank.

Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
	Continuation of	15016821	2015-02-05
Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
15016821	Continuation of	11107952	2005-04-18

PTO/AIA/14 (07-14)
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	2018-00816
	Application Number	
Title of Invention	Multimedia System For Mobile Client Platforms	

Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the **Add** button. Add

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.

Authorization to Permit Access:

☒ Authorization to Permit Access to the Instant Application by the Participating Offices

PTO/AIA/14 (07-14)
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	2018-00816
	Application Number	
Title of Invention	Multimedia System For Mobile Client Platforms	

If 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 World Intellectual Property Office (WIPO), and any other intellectual property offices in which a foreign application claiming priority to the instant patent application is filed access to the instant patent application. See 37 CFR 1.14(c) and (h). This box should not be checked if the applicant does not wish the EPO, JPO, KIPO, WIPO, or other intellectual property office in which a foreign application claiming priority to the instant patent application is filed to have access to the instant patent application.

In accordance with 37 CFR 1.14(h)(3), access will be provided to a copy of the instant patent application with respect to: 1) the instant patent application-as-filed; 2) any foreign application to which the instant patent application claims priority under 35 U.S.C. 119(a)-(d) if a copy of the foreign application that satisfies the certified copy requirement of 37 CFR 1.55 has been filed in the instant patent application; and 3) any U.S. application-as-filed from which benefit is sought in the instant patent application.

In accordance with 37 CFR 1.14(c), access may be provided to information concerning the date o f filing this Authorization.

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. ☐

Prefix	**Given Name**	Middle Name	**Family Name**	Suffix

PTO/AIA/14 (07-14)
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	2018-00816
	Application Number	
Title of Invention	Multimedia System For Mobile Client Platforms	

Mailing Address Information For Applicant:			
Address 1			
Address 2			
City		**State/Province**	
Country		Postal Code	
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 subsitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

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		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

PTO/AIA/14 (07-14)
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	2018-00816
	Application Number	
Title of Invention	Multimedia System For Mobile Client Platforms	

Signature:

Remove

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Signature	/Yasin Bismilla/			Date (YYYY-MM-DD)	2018-11-05
First Name	Yasin	Last Name	Bismilla	Registration Number	64756

Additional Signature may be generated within this form by selecting the Add button. Add

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 C o o p eration Treaty.

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.

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	Multimedia system for mobile client platforms

As the below named inventor, I hereby declare that:

This declaration is directed to:

- [■] 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: Mark Krebs Date (Optional): 3 February, 2016

Signature:

Note: An application data sheet (PTO/SB/14 or equivalent), including naming the entire inventive entity, must accompany this form or must have been previously filed. Use an additional PTO/AIA/01 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.

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 disclosure of these records is required by the Freedom of Information Act.
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 Cooperation Treaty.
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 inspection 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.

DocCode – SCORE

SCORE Placeholder Sheet for IFW Content

Application Number: 16181285 Document Date: 11/05/2018

The presence of this form in the IFW record indicates that the following document type was received in electronic format on the date identified above. This content is stored in the SCORE database.

Since this was an electronic submission, there is no physical artifact folder, no artifact folder is recorded in PALM, and no paper documents or physical media exist. The TIFF images in the IFW record were created from the original documents that are stored in SCORE.

- Drawing

At the time of document entry (noted above):

- USPTO employees may access SCORE content via eDAN using the Supplemental Content tab, or via the SCORE web page.
- External customers may access SCORE content via PAIR using the Supplemental Content tab.

Form Revision Date: August 26, 2013

Electronic Patent Application Fee Transmittal

Application Number:	16181285
Filing Date:	
Title of Invention:	Multimedia System For Mobile Client Platforms
First Named Inventor/Applicant Name:	Mark Sinclair KREBS
Filer:	Yasin Bismilla
Attorney Docket Number:	2018-00816

Filed as Micro Entity

Filing Fees for Utility under 35 USC 111(a)

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
BASIC FILING FEE-UTILITY	3011	1	75	75
UTILITY SEARCH FEE	3111	1	165	165
UTILITY EXAMINATION FEE	3311	1	190	190
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
		Total in USD ($)		**430**

Electronic Acknowledgement Receipt	
EFS ID:	34218342
Application Number:	16181285
International Application Number:	
Confirmation Number:	1045
Title of Invention:	Multimedia System For Mobile Client Platforms
First Named Inventor/Applicant Name:	Mark Sinclair KREBS
Correspondence Address:	Elan IP Inc. - 2 Bloor Street East Unit 3500 Toronto ON M4W1A8 CA 647-680-3526 USPTO@dockettrak.com
Filer:	Yasin Bismilla
Filer Authorized By:	
Attorney Docket Number:	2018-00816
Receipt Date:	05-NOV-2018
Filing Date:	
Time Stamp:	21:29:19
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Fee Worksheet (SB06)	fee-info.pdf	35046 62afa373ecc3bad08569283e327669594f507389	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	35046		

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.

<u>New Applications Under 35 U.S.C. 111</u>
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.
<u>National Stage of an International Application under 35 U.S.C. 371</u>
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.
<u>New International Application Filed with the USPTO as a Receiving Office</u>
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
16/181,285	11/05/2018	Mark Sinclair KREBS	2018-00816

CONFIRMATION NO. 1045

132285
Elan IP Inc.
2 Bloor Street East
Unit 3500
Toronto, ON M4W1A8
CANADA

FORMALITIES LETTER

OC000000104082641

Date Mailed: 11/29/2018

NOTICE TO FILE MISSING PARTS OF NONPROVISIONAL APPLICATION

FILED UNDER 37 CFR 1.53(b)

Filing Date Granted

Items Required To Avoid Abandonment:

An application number and filing date have been accorded to this application. The item(s) indicated below, however, are missing.

Applicant is given **TWO MONTHS** from the date of this Notice within which to file all required items below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

- Additional claim fees of $ **460** as a micro entity, including any required multiple dependent claim fee, are required. Applicant must submit the additional claim fees or cancel the additional claims for which fees are due.

SUMMARY OF FEES DUE:

The fee(s) required within **TWO MONTHS** from the date of this Notice to avoid abandonment is/are itemized below. Micro entity discount is in effect.

- $ **460** for **4** independent claims over 3.
- $(**0**) previous unapplied payment amount.
- $ **460** TOTAL FEE BALANCE DUE.

Replies must be received in the USPTO within the set time period or must include a proper Certificate of Mailing or Transmission under 37 CFR 1.8 with a mailing or transmission date within the set time period. For more information and a suggested format, see Form PTO/SB/92 and MPEP 512.

Replies should be mailed to:

Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

Registered users of EFS-Web may alternatively submit their reply to this notice via EFS-Web, including a copy of this Notice and selecting the document description "Applicant response to Pre-Exam Formalities Notice". https://sportal.uspto.gov/authenticate/AuthenticateUserLocalEPF.html

For more information about EFS-Web please call the USPTO Electronic Business Center at 1-866-217-9197 or visit our website at http://www.uspto.gov/ebc.

If you are not using EFS-Web to submit your reply, you must include a copy of this notice.

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

/tlulu/

Google Exhibit No. 1008
Page 110

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number: 16/181,285

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	75
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	165
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	190
TOTAL CLAIMS (37 CFR 1.16(i))	20 minus 20 =	*			OR	x 25 =	0.00
INDEPENDENT CLAIMS (37 CFR 1.16(h))	7 minus 3 =	* 4				x 115 =	460
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	890

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	SMALL ENTITY RATE($)	SMALL ENTITY ADDITIONAL FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE($)
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
16/181,285	11/05/2018	2447	430	2018-00816	20	7

CONFIRMATION NO. 1045

FILING RECEIPT

132285
Elan IP Inc.
2 Bloor Street East
Unit 3500
Toronto, ON M4W1A8
CANADA

OC000000104082640

Date Mailed: 11/29/2018

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)

Mark Sinclair KREBS, Toronto, CANADA;

Applicant(s)

Mark Sinclair KREBS, Toronto, CANADA;

Power of Attorney: None

Domestic Priority data as claimed by applicant

This application is a CON of 15/016,821 02/05/2016 *
which is a CON of 11/107,952 04/18/2005 ABN
(*)Data provided by applicant is not consistent with PTO records.

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: No

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: 11/27/2018

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 16/181,285**

Projected Publication Date: 03/07/2019

Non-Publication Request: No

Early Publication Request: No
**** MICRO ENTITY ****
Title

Multimedia System For Mobile Client Platforms

Preliminary Class

709

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.

ELAN IP

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	KREBS	Confirmation No.:	1045
Application No:	16/181,285	Art Unit:	2447
Filed:	November 5, 2018	Examiner:	TBD
For:	MULTIMEDIA SYSTEM FOR MOBILE CLIENT PLATFORMS	Attorney Docket No.:	2018-00816

PRELIMINARY AMENDMENT

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA
22313-1450

Dear Sir/Madam:

This paper is provided as a preliminary in response to the Notice to File Missing Parts mailed November 29, 2018

Amendments to the Claims begin on page 2 of this paper.

Arguments begin on page 5 of this paper.

Electronic Acknowledgement Receipt	
EFS ID:	34908309
Application Number:	16181285
International Application Number:	
Confirmation Number:	1045
Title of Invention:	Multimedia System For Mobile Client Platforms
First Named Inventor/Applicant Name:	Mark Sinclair KREBS
Customer Number:	132285
Filer:	Yasin Bismilla
Filer Authorized By:	
Attorney Docket Number:	2018-00816
Receipt Date:	20-JAN-2019
Filing Date:	05-NOV-2018
Time Stamp:	10:33:58
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Applicant Response to Pre-Exam Formalities Notice	2018-00816-preliminaryAmendment.pdf	125214 2cce03b0bb597ada45caca40b7ac4bf696d08c3c	no	5

Warnings:

Information:	
Total Files Size (in bytes):	125214

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.

LISTING OF THE CLAIMS

1. - 20. (Cancelled)

21. (New) A method for transmitting and playing multimedia on wireless players comprising:

creating a plurality of discreet continuous media objects of synchronized video and audio segments from either existing multimedia files or dynamically from multimedia streams;

directly requesting by a wireless client-based multimedia player transmission of said plurality of discreet continuous media objects;

continuously playing back received discreet continuous media objects by the multimedia player; wherein the multimedia player is configured to play continuous sequences of the received discreet continuous media objects; the multimedia player maintaining smooth and continuous playback by player algorithms which adaptively select media objects based on available wireless bandwidth conditions;

wherein the continuous media objects are maintained by content servers serving the discrete continuous media objects to wireless clients during transmission to wireless devices.

.

22. (New) The method of claim 1, wherein the method is run on a distributed network for multimedia-on-demand using a centralized content server, an indexing host, a multimedia object creator and a transcoder.

23. (New) The method of claim 1, wherein the transmission of the continuous media objects is carried out by a protocol selected from HTTP, FTP, IMAP4 and NNTP, and wherein the protocol serves files in a directory structure.

24. (New) The method of claim 1, wherein the discreet continuous media objects are created from multimedia streams by transcoding an input multimedia stream into an optimal audiovisual format at an optimal encoding rate reflecting available wireless network bandwidth and dynamically converting the input multimedia stream into the continuous media objects by

splitting the encoded stream into specified intervals, and immediately transmitting to distributed content servers which transmit the created media objects to wireless clients.

25. **(New)** The method of claim 1, wherein the discreet continuous media objects are created from existing multimedia files by transcoding and input media stream into an optimal audiovisual format at an optimal encoding rate reflecting available wireless network bandwidth, and converting the input stream into the continuous media objects by splitting the encoded stream into specified intervals.

26. **(New)** The method of claim 4, wherein the dynamically created multimedia objects are maintained by content servers serving the media objects to wireless clients.

27. **(New)** The method of claim 5, wherein the input multimedia stream is scanned after specified intervals for an I-frame, and a multimedia segment is split at the I-frame to create another media object.

28. **(New)** The method of claim 1, wherein a window of media objects for streaming transmission comprises a series of media objects which can be incremented and decremented as newly created media objects are introduced to the window or transmitted to wireless clients.

29. **(New)** The method of claim 4, wherein the continuous media objects are identified when they are identified by indexical metadata when they are created by a multimedia object creator with an Internet address that can also include:

    a transport protocol,
    a varying host URL,
    a name of the media object sequence,
    a number of objects in the sequence, and
    a media object's sequence number.

30. **(New)** The method of claim 4, wherein multimedia objects are split from multiple MPEG4 composite layer streams by scanning time intervals and splitting them at next I-frames.

31. (New) The method of claim 1, further comprising, following configuration of audio and/or video decoding components for a specific sequence of media objects, the multimedia player can delay playback until the media object buffers in wireless client memory have filled or can begin playback immediately while requesting the next multimedia object; and, wherein, the multimedia object player decision is based on a speed at which the media objects are retrieved versus a playback time of each media object, the latency of requests for media objects and/or the number of media objects that can be stored in wireless client memory.

32. (New) A method for processing the large scale distribution of multimedia content in the distributed network being managed by an indexing host server, wherein:

the indexing host registers all URLs of content servers supporting particular live multimedia object transmissions and archived sequences of multimedia objects;

remote transcoding/multimedia object creating servers provide registered updates of multimedia object sequence indices to the indexing host;

remote transcoding/multimedia object creating servers also register the sequence indices of the most recent windows of live content multimedia objects with the indexing host;

wherein content servers accept and store the most current window of live content multimedia objects or the most recent non-live archives of multimedia object sequences;

content servers transmit their multimedia directly to wireless clients, or indirectly through wireless network proxy servers; and the indexing host verifies the wireless client;

the indexing host accepts requests from wireless clients for multimedia content;

the indexing host determines the most suitable content server for the wireless client;

and, the indexing host provides the wireless client with a decryption string for the requested multimedia content.

REMARKS

Claim Status and Amendments

Claims 1-20 have been cancelled. Claims 21-32 have been added. No new subject matter has been added by way of these amendments.

Remarks

Applicant has reduced the claims on file, which now have two independent claims and a total of 12 claims. Accordingly, the additional claim fees are no longer required.

/Yasin Bismilla/
Yasin Bismilla
Reg. No.: 64756
Phone: 647.680.3526

Elan IP Inc.

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
16/181,285	11/05/2018	2425	430	2018-00816	12	2

CONFIRMATION NO. 1045

UPDATED FILING RECEIPT

132285
Elan IP Inc.
2 Bloor Street East
Unit 3500
Toronto, ON M4W1A8
CANADA



Date Mailed: 01/24/2019

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)

Mark Sinclair KREBS, Toronto, CANADA;

Applicant(s)

Mark Sinclair KREBS, Toronto, CANADA;

Power of Attorney: None

Domestic Priority data as claimed by applicant

This application is a CON of 15/016,821 02/05/2016 PAT 10171873 *
which is a CON of 11/107,952 04/18/2005 ABN
(*)Data provided by applicant is not consistent with PTO records.

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: No

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: 11/27/2018

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 16/181,285**

Projected Publication Date: 03/07/2019

Non-Publication Request: No

Early Publication Request: No
**** MICRO ENTITY ****
Title

Multimedia System For Mobile Client Platforms

Preliminary Class

725

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.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number: 16/181,285

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	75
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	165
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	190
TOTAL CLAIMS (37 CFR 1.16(i))	12 minus 20 =	*			OR	x 25 =	0.00
INDEPENDENT CLAIMS (37 CFR 1.16(h))	2 minus 3 =	*				x 115 =	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	430

APPLICATION AS AMENDED - PART II

AMENDMENT A

	(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($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE($)
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	

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE($)	ADDITIONAL FEE($)		RATE($)	ADDITIONAL FEE($)
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	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
16/181,285	11/05/2018	Mark Sinclair KREBS	2018-00816

CONFIRMATION NO. 1045

132285
Elan IP Inc.
2 Bloor Street East
Unit 3500
Toronto, ON M4W1A8
CANADA

PUBLICATION NOTICE

OC00000106407168

Title: Multimedia System For Mobile Client Platforms

Publication No.US-2019-0075361-A1
Publication Date:03/07/2019

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

PTO/SB/06 (09-11)
Approved for use through 1/31/2014. 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.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number 16/181,285	Filing Date 11/05/2018	☐ To be Mailed

ENTITY: ☑ LARGE ☐ SMALL ☐ MICRO

APPLICATION AS FILED - PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA		RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A		N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A		N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A		N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*		x $100 =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*		x $460 =	
☐ 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).				
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))					
* If the difference in column 1 is less than zero, enter "0" in column 2.				TOTAL	

APPLICATION AS AMENDED - PART II

		(Column 1)		(Column 2)	(Column 3)			
AMENDMENT	09/13/2019	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA		RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 8	Minus	** 20	= 0		x $100 =	0
	Independent (37 CFR 1.16(h))	* 3	Minus	*** 3	= 0		x $460 =	0
	☐ Application Size Fee (37 CFR 1.16(s))							
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							
							TOTAL ADD'L FEE	0

		(Column 1)		(Column 2)	(Column 3)			
AMENDMENT		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA		RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	*	Minus	**	=		x $0 =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=		x $0 =	
	☐ Application Size Fee (37 CFR 1.16(s))							
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							
							TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.	LIE
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".	/FREDERICK E BRISCOE/
*** 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 number found in the appropriate box in column 1.	

This collection of information is required by 37 CFR 1.16. 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 12 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.

Electronic Acknowledgement Receipt	
EFS ID:	37160875
Application Number:	16181285
International Application Number:	
Confirmation Number:	1045
Title of Invention:	Multimedia System For Mobile Client Platforms
First Named Inventor/Applicant Name:	Mark Sinclair KREBS
Customer Number:	132285
Filer:	Yasin Bismilla
Filer Authorized By:	
Attorney Docket Number:	2018-00816
Receipt Date:	13-SEP-2019
Filing Date:	05-NOV-2018
Time Stamp:	15:03:53
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Preliminary Amendment	2018-00816-preliminaryAmendment2.pdf	150320 22a7ab87d297e482813d12b699416e359f169233	no	5
Warnings:					

Information:	
Total Files Size (in bytes):	150320

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.

ELAN IP

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	KREBS	Confirmation No.:	1045
Application No:	16/181,285	Art Unit:	2447
Filed:	November 5, 2018	Examiner:	TBD
For:	MULTIMEDIA SYSTEM FOR MOBILE CLIENT PLATFORMS	Attorney Docket No.:	2018-00816

PRELIMINARY AMENDMENT

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA
22313-1450

Dear Sir/Madam:

This paper is presented as a pre-examination preliminary amendment to the claims.

Amendments to the Claims begin on page 2 of this paper.

Arguments begin on page 5 of this paper.

LISTING OF THE CLAIMS

1. – 32 (Cancelled)

33. (New) A method comprising:

receiving audio and video segments encoded in a digital encoding format and with an encoding rate;

wherein, said audio and video segments include metadata to form multimedia objects;

requesting by a multimedia player, transmission of said multimedia objects;

wherein said multimedia objects are located using http and received by said multimedia player from servers using a wireless connection;

playing back received multimedia objects by the multimedia player, wherein the multimedia player is configured to play multimedia objects in a sequence such that fluidity, video quality and audio quality are maintained by selecting a plurality of said multimedia objects that reflect available network bandwidth, adjusting said selection and playback according to the multimedia object metadata, and by utilizing optimized decoding processes to maintain quality playback.

34. (New) The method of claim 33, wherein the transmission of the multimedia objects is carried out by protocols that can serve files in a directory.

35. (New) The method of claim 33, wherein the multimedia objects are created from a live multimedia stream that is transcoded into an optimal digital format at an optimal encoding rate reflecting available network bandwidth and converting the input multimedia stream into audio and video segments by splitting the encoded stream into intervals for access by wireless clients.4. The method of claim 1, wherein the discrete multimedia objects are created from existing multimedia files that have been transcoded into optimal digital formats at an optimal encoding rates reflecting available network bandwidth, and converting them into audio and video segments by splitting the encoded files into intervals for access by wireless clients.

36. (New) The method of claim 33, wherein a multimedia file is scanned for an I-frame, and a multimedia segment is split at the I-frame to create the video segment, together with a separate audio segment.

Google Exhibit No. 1008
Page 131

37. (New) The method of claim 33, wherein the multimedia objects are identified when they are created by a multimedia object creator with:

a host URL and path multimedia object's sequence number or a number indicating its position within the multimedia object sequence;
and metadata,

wherein metadata is used to configure the multimedia player for audio and video playback.

38. (New) The method of claim 33, wherein multimedia objects are split from multiple MPEG4 composite layer streams by scanning time intervals and splitting the multimedia objects into video multimedia objects at next I-frames.

39. (New) A multimedia player comprising:

computer executable instructions stored on a non-transitory computer readable medium which when executed by a processor:

request transmission of multimedia objects;

receive audio and video segments encoded in a digital encoding format and with an encoding rate; wherein, said audio and video segments include metadata to form multimedia objects; and wherein said multimedia objects are located using http and received by said multimedia player from servers using a wireless connection;

play back received multimedia objects in a sequence such that fluidity, video quality and audio quality are maintained by selecting a plurality of said multimedia objects that reflect available network bandwidth, adjusting said selection and playback according to the multimedia object metadata, and by utilizing optimized decoding processes to maintain quality playback.

40. (New) A system comprising

at least one centralized content server transmitting multimedia objects; said multimedia objects including segmented audio and video, and metadata;

at least one multimedia player configured to:

request transmission of multimedia objects;

receive audio and video segments encoded in a digital encoding format and with an encoding rate; wherein, said audio and video segments include metadata to form multimedia objects; and wherein said multimedia objects are located using http and received by said multimedia player from servers using a wireless connection;

play back received multimedia objects in a sequence such that fluidity, video quality and audio quality are maintained by selecting a plurality of said multimedia objects that reflect available network bandwidth, adjusting said selection and playback according to the multimedia object metadata, and by utilizing optimized decoding processes to maintain quality playback.

REMARKS

Claim Status and Amendments

Claims 1-32 have been cancelled. Claims 21-32 have been added. No new subject matter has been added by way of these amendments, which are entered for clarity.

.

/Yasin Bismilla/
Yasin Bismilla
Reg. No.: 64756
Phone: 647.680.3526

Elan IP Inc.

Doc code : PET.OP.AGE
Description : Petition to make special based on Age/Health

PTO/SB/130 (07-09)

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

PETITION TO MAKE SPECIAL BASED ON AGE FOR ADVANCEMENT OF EXAMINATION UNDER 37 CFR 1.102(c)(1)

Application Information

Application Number	16181285	Confirmation Number	1322	Filing Date	05-Nov-2018
Attorney Docket Number (optional)	2018-00816	Art Unit	2486	Examiner	KRISTIN DOBBS
First Named Inventor	Mark Sinclair KREBS				
Title of Invention	Multimedia System For Mobile Client Platforms				

Attention: Office of Petitions

An application may be made special for advancement of examination upon filing of a petition showing that the applicant is 65 years of age, or more. No fee is required with such a petition. See 37 CFR 1.102(c)(1) and MPEP 708.02 (II).

APPLICANT HEREBY PETITIONS TO MAKE SPECIAL FOR ADVANCEMENT OF EXAMINATION IN THIS APPLICATION UNDER 37 CFR 1.102(c)(1) and MPEP 708.02 (II) ON THE BASIS OF THE APPLICANT'S AGE.

A grantable petition requires one of the following items:
(1) Statement by one named inventor in the application that he/she is 65 years of age, or more; or
(2) Certification by a registered attorney/agent having evidence such as a birth certificate, passport, driver's license, etc. showing one named inventor in the application is 65 years of age, or more.

Name of Inventor who is 65 years of age, or older

Given Name	Middle Name	Family Name	Suffix
Mark	Sinclair	KREBS	

A signature of the applicant or representative is required in accordance with 37 CFR 1.33 and 10.18.
Please see 37 CFR 1.4(d) for the format of the signature.

Select (1) or (2) :

○ (1) I am an inventor in this application and I am 65 years of age, or more.

◉ (2) I am an attorney or agent registered to practice before the Patent and Trademark Office, and I certify that I am in possession of evidence, and will retain such in the application file record, showing that the inventor listed above is 65 years of age, or more.

Signature	/Yasin Bismilla/		
Name	Yasin Bismilla	Registration Number	64756

EFSWeb 1.0.18

PTO/SB/130 (07-09)
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

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 Fr eedom 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 Cooperation Treaty.

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 indivi duals.

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.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

In re Application of
Mark Sinclair KREBS

Application No. 16181285
Filed: November 5,2018

Attorney Docket No. 2018-00816

:
:
:DECISION ON PETITION TO MAKE SPECIAL
:UNDER 37 CFR 1.102(c)(1)
:

This is a decision on the electronic petition under 37 CFR 1.102 (c)(1), filed 13-SEP-2019 to make the above-identified application special based on applicant's age as set forth in MPEP § 708.02, Section II.

The petition is **GRANTED**.

A grantable petition to make an application special under 37 CFR 1.102(c)(1), MPEP § 708.02, Section II : Applicant's Age must include a statement by applicant or a registered practitioner having evidence that applicant is at least 65 years of age. No fee is required.

Accordingly, the above-identified application has been accorded "special" status and will be taken up for action by the examiner upon the completion of all pre-examination processing.

Telephone inquiries concerning this electronic decision should be directed to the Electronic Business Center at 866-217-9197.

All other inquiries concerning either the examination or status of the application should be directed to the Technology Center.

Electronic Acknowledgement Receipt

EFS ID:	37160959
Application Number:	16181285
International Application Number:	
Confirmation Number:	1045
Title of Invention:	Multimedia System For Mobile Client Platforms
First Named Inventor/Applicant Name:	Mark Sinclair KREBS
Customer Number:	132285
Filer:	Yasin Bismilla
Filer Authorized By:	
Attorney Docket Number:	2018-00816
Receipt Date:	13-SEP-2019
Filing Date:	05-NOV-2018
Time Stamp:	15:06:54
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Petition automatically granted by EFS	petition-request.pdf	38180 ba5fa1976d5de081ff7ab2c18b2ab1abc9171aa8	no	2

Warnings:

Information:	
Total Files Size (in bytes):	38180

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/SB/08a (01-10)
Approved for use through 07/31/2012. OMB 0651-0031
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.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT
(Not for submission under 37 CFR 1.99)

Application Number	16181285
Filing Date	2018-11-05
First Named Inventor	Mark Sinclair KREBS
Art Unit	2486
Examiner Name	DOBBS, KRISTIN SENSMEIER
Attorney Docket Number	2018-00816

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	5202961	A	1993-04-13	Mills et al.	
	2	5414455	A	1995-05-09	Hooper et al.	
	3	5434678	A	1995-07-18	Abecassis	
	4	5434860	A	1995-07-18	Riddle	
	5	5579239	A	1996-11-26	Freeman et al.	
	6	5610841	A	1997-03-11	Tanaka et al.	
	7	5612742	A	1997-03-18	Krause et al.	
	8	5751883	A	1998-05-12	Ottesen et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	16181285
	Filing Date	2018-11-05
	First Named Inventor	Mark Sinclair KREBS
	Art Unit	2486
	Examiner Name	DOBBS, KRISTIN SENSMEIER
	Attorney Docket Number	2018-00816

	9	5751968	A	1998-05-12	Cohen	
	10	5819160	A	1998-10-06	Foladare et al.	
	11	5821986	A	1998-10-13	Yuan et al.	
	12	5841432	A	1998-11-24	Carmel et al.	
	13	5892915	A	1999-04-06	Duso et al.	
	14	5903673	A	1999-05-11	Wang et al.	
	15	5928330	A	1999-07-27	Goetz et al.	
	16	5933603	A	1999-08-03	Vahalia et al.	
	17	5953506	A	1999-09-14	Kalra et al.	
	18	5974503	A	1999-10-26	Venkatesh et al.	
	19	5987510	A	1999-11-16	Imai et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	16181285
	Filing Date	2018-11-05
	First Named Inventor	Mark Sinclair KREBS
	Art Unit	2486
	Examiner Name	DOBBS, KRISTIN SENSMEIER
	Attorney Docket Number	2018-00816

	20	7295608	B2	2007-11-13	Reynolds et al.	
	21	6490320	B1	2002-12-03	Vetro et al.	
	22	7120194	B2	2006-10-10	Lee	
	23	5996015	A	1999-11-30	Day et al.	
	24	6014694	A	2000-01-11	Aharoni et al.	
	25	6014706	A	2000-01-11	Cannon et al.	
	26	6032193	A	2000-02-29	Sullivan	
	27	6112226	A	2000-08-29	Weaver et al.	
	28	6119154	A	2000-09-12	Weaver et al.	
	29	6138147	A	2000-10-24	Weaver et al.	
	30	6151632	A	2000-11-21	Chaddha et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	16181285
	Filing Date	2018-11-05
	First Named Inventor	Mark Sinclair KREBS
	Art Unit	2486
	Examiner Name	DOBBS, KRISTIN SENSMEIER
	Attorney Docket Number	2018-00816

	31	6195352	B1	2001-02-27	Cushman et al.	
	32	6292834	B1	2001-09-18	Ravi et al.	
	33	6385596	B1	2002-05-07	Wiser et al.	
	34	6397230	B1	2002-05-28	Carmel et al.	
	35	6536043	B1	2003-03-18	Guedalia	
	36	6389473	B1	2002-05-14	Carmel et al.	

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	20040194144	A1	2004-09-30	Lin et al.	
	2	20020120675	A1	2002-08-29	Everett et al.	
	3	20070298719	A1	2007-12-27	Sproule et al.	

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	16181285
	Filing Date	2018-11-05
	First Named Inventor	Mark Sinclair KREBS
	Art Unit	2486
	Examiner Name	DOBBS, KRISTIN SENSMEIER
	Attorney Docket Number	2018-00816

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] i	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1	9730551	WO	A1	1997-08-21	Olivr Corp Ltd.		
	2	0762300	EP	A2	1997-03-12	Toshiba KK		

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]
	1	ALEX ZAMBELLI, "IIS Smooth Streaming Technical Overview," Microsoft Silverlight Windows Server Internet Information Services, March 2009, pgs. 1-17, Microsoft Corporation	
	2	LI ET AL., "Two Decades of Internet Video Streaming: A Retrospective View", ACM Transactions on Multimedia Computing, October, 2013, pgs 1-20, Vol. 33 No. 9	
	3	LUIZ FERNANDO CAPRETZ, "A Brief History of the Object-Oriented Approach," ACM SIGSOFT, March 2003, Vol. 28. No. 6., pgs. 1-11, Software Engineering Notes, 10.1145/638750.638778	
	4	SCHWARTZ COMMUNICATIONS, INC., "GEO Interactive Unveils Emblaze Live Technology; Revolutionary Emblaze Technology Used for White House EasterLink Cybercast; No plug-ins or Server Software Required," The Free Library, April 13, 1998, pgs. 1-3, Business Wire	
	5	REALNETWORKS INC., "RealSystem G2 Production Guide" BETA 1 Release, May 14 2002, pgs. 1-130	

If you wish to add additional non-patent literature document citation information please click the Add button Add

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	16181285
	Filing Date	2018-11-05
	First Named Inventor	Mark Sinclair KREBS
	Art Unit	2486
	Examiner Name	DOBBS, KRISTIN SENSMEIER
	Attorney Docket Number	2018-00816

EXAMINER SIGNATURE			
Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	16181285
	Filing Date	2018-11-05
	First Named Inventor	Mark Sinclair KREBS
	Art Unit	2486
	Examiner Name	DOBBS, KRISTIN SENSMEIER
	Attorney Docket Number	2018-00816

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

See attached certification statement.

The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☒ A certification statement is not submitted herewith.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Yasin Bismilla/	Date (YYYY-MM-DD)	2019-09-13
Name/Print	Yasin Bismilla	Registration Number	64,756

This collection of information is required by 37 CFR 1.97 and 1.98. 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 1 hour 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.**

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 record s.

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 Cooperation Treaty.

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.

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04N 7/14 — A1

(11) International Publication Number: **WO 97/30551**

(43) International Publication Date: 21 August 1997 (21.08.97)

(21) International Application Number: PCT/IL97/00055

(22) International Filing Date: 13 February 1997 (13.02.97)

(30) Priority Data:

117133	14 February 1996 (14.02.96)	IL
119655	20 November 1996 (20.11.96)	IL
08/788,830	6 January 1997 (06.01.97)	US

(71) Applicant *(for all designated States except US)*: OLIVR CORPORATION LTD. [IL/IL]; Kanfei Nesharim Street 24, 95464 Jerusalem (IL).

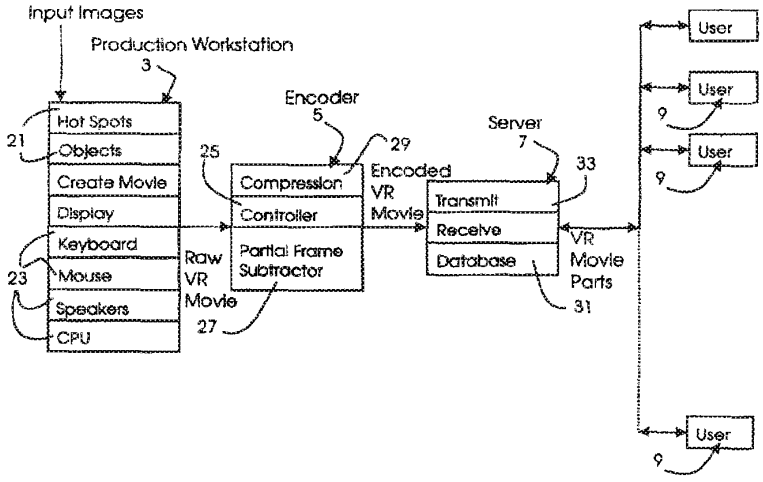
(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: GUEDALIA, Jacob, Leon [IL/IL]; Haporzim Street 24, 92541 Jerusalem (IL).

(74) Agents: COLB, Sanford, T. et al.; Sanford T. Colb & Co., P.O. Box 2273, 76122 Rehovot (IL).

(81) Designated States: AL, AM, AT, AT (Utility model), AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, CZ (Utility model), DE, DE (Utility model), DK, DK (Utility model), EE, EE (Utility model), ES, FI, FI (Utility model), GB, GE, HU, IL, 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, SK (Utility model), TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ARIPO patent (KE, LS, MW, SD, SZ, UG), 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: METHOD AND SYSTEMS FOR PROGRESSIVE ASYNCHRONOUS TRANSMISSION OF MULTIMEDIA DATA



(57) Abstract

A system for transmitting digital data representing an original over plural transmission links (101), at least some of which have limited bandwidth, including a digital data source storing digital data representing the original, a digital data receiver receiving data representing the original via one of the plural transmission links having limited bandwidth and a digital transmitter for transmitting the digital data representing the original to the receiver over a transmitting link having a limited bandwidth in plural blocks (76) which are sequentially transmitted at a rate determined by the limited bandwidth, each block being an incomplete collection of data which includes parts of multiple frames (72), receipt of subsequent blocks by the receiver being used to cumulatively improve the quality of the digital data viewed by the receiver.

Google Exhibit No. 1008

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.

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

METHOD AND SYSTEMS FOR PROGRESSIVE ASYNCHRONOUS TRANSMISSION OF MULTIMEDIA DATA

FIELD OF THE INVENTION

The present invention relates to methods and systems for encoding digital multimedia data for transmission over a network.

BACKGROUND OF THE INVENTION

The following U.S. Patents have been found in a U.S. Patent Search and are believed to be generally relevant to the field of the invention:

4,897,867	1/90	Foster et al.
5,119,188	6/92	McCalley et al.
5,122,873	6/92	Golin
5,195,092	3/93	Wilson et al.
5,220,420	6/93	Hoarty et al.
5,236,199	8/93	Thompson, Jr.
5,251,209	10/93	Jurkevich et al.
5,265,248	11/93	Moulios et al.
5,283,819	1/94	Glick et al.
5,325,423	6/94	Lewis
5,351,276	9/94	Doll, Jr. et al.
5,363,482	11/94	Victor et al.
5,420,572	5/95	Dolin, Jr. et al.
5,420,801	5/95	Dockter et al.
5,438,658	8/95	Fitzpatrick et al.
5,487,167	1/96	Dinallo et al.
5,495,576	2/96	Ritchey
5,508,940	4/96	Rossmere et al.
5,519,435	5/96	Anderson
5,553,221	9/96	Reimer et al.

Google Exhibit No. 1008
Page 150

5,553,222	9/96	Milne et al.
5,557,538	9/96	Retter et al.
5,561,791	10/96	Mendelson et al.
5,564,001	10/96	Lewis
5,577,180	11/96	Reed
5,577,258	11/96	Cruz et al.
5,581,783	12/96	Ohashi

When using various media such as video, audio, text and images, a user generally retrieves the media from a storage device or "server" connected via a network to many computers or users. The server downloads the media to the network and transmits it to the user at the user's request.

There are two basic limitations involved in such data retrieval: delay between the time that a user requests the data and the time when the server downloads it to the network, and bandwidth limitations on data throughput and rate of data transmission. The present invention relates to the second limitation.

One example of such a system includes a CD ROM drive and personal computer which may be located at the same site. Another example includes a network connecting Internet servers and users' personal computers. Such networks are installed in order to facilitate convenient data transmission between users and data distribution from the server to the users' computers.

When a user retrieves the media from a storage device or server, typically connected via a network, the aforesaid bandwidth limitations affect the amount of time required to transmit a video frame from the server to the user, and thus limit the video frame rate. Moreover, when dealing with object movies and panoramas the files being transmitted are extremely large, so that overcoming bandwidth limitations is a critical enabling factor, even for high bandwidth networks.

Currently, two methods are employed to overcome bandwidth limitations: The first is to compress the video frame sequence, thereby speeding up transmission time at the cost of additional downstream processing to decompress the frames prior to display. The second is to copy the entire sequence to an intermediate storage device, such as a user's hard disk, to which the user has higher bandwidth access, at the cost of delaying the viewing of the video until the entire sequence has been delivered.

Google Exhibit No. 1008
Page 151

Known network applications involve streaming data from a server to a client computer (hereinafter also referred to as "client"). "Streaming" refers to serial or parallel transmission of digital data between two computers, by transmitting sequences of bit packets. For example, installation executables on a network server stream files to a client computer performing the installation. Servers with large amounts of memory are used to archive digital movies, which are streamed to a client computer for viewing upon demand. Digital video is broadcast from cable stations to subscribers using streaming. Internet browsers, such as Netscape and Microsoft Explorer, are used to stream data from a server on the web to a client. Internet web sites can contain enormous databases, such as phone directories for all of the cities in the U.S., photographs from art galleries and museums around the world, voluminous encyclopedias, and even copies of all patents ever issued by the U.S. Patent & Trademark Office. Clients using the Internet can search these databases and then request the server to download specific information. This request initiates a streaming event.

In view of the multitude of bandwidths present in complex client/server systems today and the large amounts of data necessary to produce compelling audio, video and imaging presentations, there is a pressing demand for scalable data representation of multimedia data, so that it can be delivered for on-line interactive playback in such a form that the transmission rate can accommodate the client bandwidth. The Internet is an example of a complex system where clients with many different bandwidth constraints demand multimedia data from server sites. There are several low bandwidth rates for modem transmission over phone lines, higher rates for frame relay lines, higher rates for ISDN lines, even higher rates for T1 lines, etc.

Google Exhibit No. 1008
Page 152

SUMMARY OF THE INVENTION

The present invention seeks to provide an improved method and system for transmitting digital data representing the original over plural transmission links at least some of which have limited bandwidth.

The present invention relates to scalable encoding, which enables two or more clients, connected to a server by lines having differing bandwidth, to begin playing the multimedia data on-line, and both at the same time, almost immediately after the start of streaming, but the lower bandwidth client receives lower quality media than the higher bandwidth client at first. As the media is replayed in the foreground and the bandwidth is freed, more data streams in via background, and the quality of the media is enhanced.

Without scalable encoding, a client of an Internet application must wait until the requested data arrives, at whatever rate its network line provides. A client with a 14.4 Kbs modem line, for example, would have to wait twice as long as a client with a 28.8 Kbs modem line. Moreover, if the data was encoded for 28.8 Kbs on-line playback, the 14.4 Kbs client would never be able to achieve live playback, since there would be an ever-increasing lag in the data stream. Conversely, if the data were encoded for 14.4 Kbs on-line playback, the 28.8 Kbs client would receive unnecessarily poor quality media.

On account of the progressive nature of the data representation in accordance with the present invention, the additional data block arriving in a 14.4 Kbs stream combines with the previous data block which arrived in a 14.4 Kbs stream, to produce a 28.8 Kbs streamed version; all that is being sent is the incremental data necessary for the upgrade. The progressive form of the encoding itself provides the ability to achieve scalability.

Another shortcoming of non-scalable encoding as in the prior art is the inability to preview a video sequence. Often a client would like to play a quick preview of a video clip, before deciding whether or not to download it. The scalable representation of the present invention can be used to deliver the video in a preview mode, as the first data blocks. If the client continues to download the video after previewing, the first data

Google Exhibit No. 1008
Page 153

block already transmitted is progressively integrated with additional data blocks to create the full viewing video.

The present invention can also be applied to enhance delivery of large still images for multi-resolution gazing. Current technology transmits such images as large files, and carries out extensive computations for sub-sampling to lower resolution and zooming in to areas of interest for gazing. This makes it very time consuming to interact with large images, and as a result it is currently impractical to produce high resolution images for Internet browsing. When dealing with large images, producers simply sub-sample them to fit entirely within a computer monitor screen, and store the resulting low resolution images on web servers. Using the technology of the present invention, producers can deliver high resolution images over the Internet for rapid interactive gazing.

The present invention seeks to provide a scalable representation of multimedia data, enabling the data to be (a) progressively streamed, (b) transmitted asynchronously to clients at different bandwidths and (c) played back interactively on-line. The representation is two-dimensional, with one dimension (block number) being characterized by progressiveness in quality, and the second dimension (frame number) being characterized by interactivity.

The representation comprises data blocks which are integrated with one another to produce successively higher bandwidth versions of the media, the data blocks comprising encoded frames. The first data block corresponds to the lowest bandwidth, and enables the client with this bandwidth to play back the media on-line at the lowest quality. The second data block, when integrated with the first block, corresponds to the next higher bandwidth, and enables the client with this bandwidth to play back the media on-line at the next highest quality, and similarly for each successive data block.

Moreover, a client with the lowest bandwidth who played the media at the low quality and freed the bandwidth can continue in background to receive successive data blocks and integrate them with previously received data blocks, resulting in successively higher quality media each time it is replayed. The modular form of the data representation thus makes it possible to both accommodate different bandwidths and progressively update media quality. A production tool makes it possible for a producer to control modularity and quality settings.

Google Exhibit No. 1008
Page 154

In accordance with the present invention there is provided a method for providing on-line virtual reality movies, comprising inputting a cyclic movie sequence into an encoder, determining the number of portions that each frame of said movie is divided into, and forming partial frames, specifying hot-spots and independent objects for interaction within a partial frame, transmitting the partial frames part by part to a user's asynchronous database, and displaying said frames on a user's interface.

There is also provided in accordance with a preferred embodiment of the present invention a system for producing virtual reality (VR) movies comprising an encoder for preparing the VR movie for transmission, and a server including a repository for the VR movie and a transceiver for transmitting the movie, part by part to a user, upon request.

An essential feature of the present invention is the use of a two-dimensional interactive progressive database to represent multimedia data, and the storage of this database in three different forms: for streaming, processing and playback purposes. The database is calibrated in data blocks of roughly equal size, to deliver the media for on-line playback at a selected range of bandwidths, and in such a way that the higher bandwidth versions are built by integrating data blocks with the lower bandwidth versions. Thus, rather than discard the lower bandwidth data, it is saved and used directly to upgrade from low to high bandwidth quality.

The data blocks themselves are comprised of frames which can be randomly accessed, thus giving a second dimension (namely, frames) to the progressive database. Thus it is possible to selectively build higher quality versions of some frames and not others. The mechanism determining which frames to send within each block may be controlled interactively by the user.

To best appreciate the achievement of the present invention, one should consider it in the broader perspective of streaming technology. There are known two modes of accessing streamed data: random access and sequential access. In a random access mode, the client can request the server to transmit data from any position within the stream. In a sequential access mode, the server can only send a serial stream.

Current Internet streaming technology does not provide random access. Rather, the stream is sent serially in packets. The present invention operates by creating three copies of the progressive database. A first copy is stored on the server serially. A

Google Exhibit No. 1008
Page 155

second copy, which mirrors the server database, is built on the client, with random accessibility. These first two copies are in encoded form. The frames in the data blocks from the second copy are decoded and stored in the third copy. The third copy is dynamically updated and contains the frames to be displayed in either raw bitmap form, or in intermediate compressed form whereby the decompression is fast enough to keep up with real time interactive display in response to user commands.

A by-product of the present invention, when applied to video clips, is the ability to deliver a preview of the video using the first data blocks. This enables the client to play the preview almost immediately after the transmission begins, and then to quickly decide whether or not to proceed with the download. Moreover, if the client does continue with the download, then the first data block already transmitted is integrated with additional data blocks being downloaded, to form the full view version of the video. Thus, rather than discard the data transmitted for the preview, it is saved and used to create the full view frame sequence.

The present invention can also be applied to efficiently deliver large still images at multiple resolutions for interactive gazing. Each block of the progressive database stores various tiles of the image at different resolutions. Smaller tiles are stored at higher resolution. Hot spots are used to link tiles at lower resolution to smaller tiles contained within them at higher resolution. When a viewer clicks on a hot spot to gaze, the display quickly brings up the tile linked to by the hot spot, giving the effect of an instant zoom in. The totality of multi-resolution tiles may comprise the "frames" in this application, and these frames form the interactivity dimension of the database.

The first tile consists of the lowest resolution version of the full image, with hot spots encoded within it. As soon as the first tile is delivered and decoded, the user can at once begin gazing at the higher resolution tiles, even though the image is of low quality. While the user is viewing the tiles, additional data blocks are being delivered and decoded in background, and the quality of the tiles is being upgraded as time progresses. In other words, the zoomed in portion of the image being gazed at gets displayed almost immediately after the streaming begins, but at low quality. The quality improves with time.

Google Exhibit No. 1008
Page 156

The higher resolution tiles correspond to hierarchical "areas of interest" in the image. The choice of which areas to mark with hot spots as areas of interest is in the hands of the producer. This use of the invention is particularly efficient in the case where there is a relatively small number of areas of interest in the full image, so that relatively few tiles are encoded at the higher resolutions. Without the present invention, the client would have to wait to receive the full image at higher resolution before viewing any part of it at this resolution, even though only small parts of it are of interest. Moreover, each zoom in and out would be both processor and memory demanding. Viewer interactivity would be painfully slow.

There is thus provided in accordance with a preferred embodiment of the present invention a system for transmitting digital data representing the original over plural transmission links at least some of which have limited bandwidth including:

a digital data source storing digital data representing an original;

a digital data receiver receiving the digital data representing an original via one of the plural transmission links having limited bandwidth; and

a digital data transmitter operative to transmit the digital data representing an original to the receiver over a transmission link having a limited bandwidth in plural blocks which are sequentially transmitted at a rate determined by the limited bandwidth, each block being an incomplete collection of data which includes parts of multiple frames, each frame being viewable in a selectable order by the receiver even when less than all of the plural blocks have been received, receipt of subsequent blocks by the receiver being used to cumulatively improve the quality of the digital data viewed by the receiver.

There is additionally provided in accordance with a preferred embodiment of the present invention a digital data transmitter actuator comprising:

an organizer operative, when actuated, to access digital data representing an original which is organized in plural blocks for subsequent transmission, each block being an incomplete collection of data which includes parts of multiple frames, each frame being viewable in a selectable order by the a receiver even when less than all of the plural blocks have been received; and

Google Exhibit No. 1008
Page 157

a receiver instruction interface responsive to interactive inputs from a receiver for actuating the organizer to select a given block and at least one given partial frame within the given block for transmission.

There is also provided in accordance with a preferred embodiment of the present invention a digital data receiver including:

a data receipt interface receiving digital data representing an original in a plurality of sequential blocks, each block being an incomplete collection of data which includes parts of multiple frames;

a block accumulator for combining plural blocks as they are received for viewing by the recipient; and

a viewer including a recipient interface which permits each frame to be viewed in an order selected by the recipient, even when less than all of the plural blocks have been received, combining of plural blocks by the block accumulator being used to improve the quality of the digital data viewed by the recipient.

There is additionally provided in accordance with a preferred embodiment of the present invention a method for transmitting digital data representing an original over plural transmission links at least some of which have limited bandwidth including the steps of:

storing digital data representing an original;

receiving at a receiver the digital data representing an original via one of the plural transmission links having limited bandwidth; and

transmitting the digital data representing an original to the receiver over a transmission link having a limited bandwidth in plural blocks which are sequentially transmitted at a rate determined by the limited bandwidth, each block being an incomplete collection of data which includes parts of multiple frames, each frame being viewable in a selectable order by the receiver even when less than all of the plural blocks have been received, receipt of subsequent blocks by the receiver being used to cumulatively improve the quality of the digital data viewed by the receiver.

There is also provided in accordance with a preferred embodiment of the present invention a method for digital data transmission including:

Google Exhibit No. 1008
Page 158

organizing digital data representing an original into plural blocks for subsequent transmission, each block being an incomplete collection of data which includes parts of multiple frames, each frame being viewable in a selectable order by the a receiver even when less than all of the plural blocks have been received;

responsive to interactive inputs from a receiver for actuating the organizer, selecting a given block and at least one given partial frame within the given block for transmission; and

transmitting the selected given block and at least one given partial frame to a user.

In accordance with a preferred embodiment of the present invention, the block accumulator is operative to combine plural blocks which are distinguished from each other by their respective frequency bands.

Preferably, the digital data receiver includes a fractal decompression engine.

In accordance with a preferred embodiment of the present invention, the data receipt interface is operative to initially receive a first plurality of blocks containing relatively low frequency data and thereafter receive a second plurality of blocks containing relatively high frequency data and the block accumulator is operative to reconstitute the digital data representing an original from the blocks representing relatively high frequency and relatively low frequency data.

Preferably, the block accumulator is operative to combine plural blocks having different sampling.

In accordance with a preferred embodiment of the present invention, the sampling rate of a combined plurality of blocks is equal to the sum of the sampling rates of individual ones of the plurality of blocks.

Preferably, the digital data receiver includes a wavelet decoder.

In accordance with a preferred embodiment of the present invention, the block accumulator includes a dequantizer which combines blocks each of which contain quantized data of a different order, such that accumulation of multiple blocks provides combined data of greater precision than that contained in any single block.

There is additionally provided in accordance with a preferred embodiment of the present invention a digital data transmitter actuator including:

Google Exhibit No. 1008
Page 159

an organizer operative, when actuated, to access digital data representing an original which is organized in plural blocks for subsequent transmission, each block being an incomplete collection of data which includes parts of multiple frames, each frame being viewable in a selectable order by the a receiver even when less than all of the plural blocks have been received; and

a receiver instruction interface responsive to interactive inputs from a receiver for actuating the organizer to select a given block and at least one given partial frame within the given block for transmission.

In accordance with a preferred embodiment of the present invention a first one of the plural blocks contains digital data which represents a first approximation to the original.

Preferably, additional ones of the plural blocks, when combined with the first one of the plural blocks provide additionally accurate approximations to the original.

In accordance with a preferred embodiment of the present invention each of the multiple frames includes a portion of data which can be independently and interactively manipulated.

Preferably, the system also includes a block generator operative to receive digital data representing the original and to provide the plural blocks.

There is also provided in accordance with a preferred embodiment of the present invention a block generator including:

a producer interface; and

a digital data compressor, operative in response to producer control parameters received via the producer interface for receiving digital data representing an original and providing plural blocks, each block being an incomplete collection of data which includes parts of multiple frames.

In accordance with a preferred embodiment of the present invention the block generator is operative to provide plural blocks which are distinguished from each other by their respective frequency bands.

Preferably, the block generator includes a fractal compression engine.

In accordance with a preferred embodiment of the present invention, the block generator is operative to decompose the digital data representing an original in relatively

Google Exhibit No. 1008
Page 160

high frequency and relatively low frequency digital data portions, and wherein a first plurality of blocks containing the relatively low frequency portion is transmitted by the data transmitter prior to transmission of a second plurality of blocks containing the relatively high frequency portion.

Preferably, the block generator is operative to provide plural blocks by sampling the received digital data.

In accordance with a preferred embodiment of the present invention the sampling rate of a plurality of blocks is equal to the sum of the sampling rates of individual ones of the plurality of blocks.

Preferably, the block generator includes a wavelet encoder.

In accordance with a preferred embodiment of the present invention, the block generator includes a quantizer which produces blocks each of which contain quantized data of a different order, such that accumulation of multiple blocks provides combined data of greater precision than that contained in any single block.

There is also provided in accordance with another preferred embodiment of the present invention, a method for encoding original digital video data to be stored on a server computer for on-line delivery to client computers, including the steps of:

encoding the digital video into a database including a series of encoded data blocks, each block including a sequence of encoded frames, with the property that successive blocks when decoded and integrated together provide successively higher bandwidth versions of the video for on-line playback;

storing the database on a server computer;

processing a request by a client computer for on-line delivery of the video in order to determine which data blocks to transmit, so as to accommodate the client bandwidth;

transmitting the necessary data blocks to the client;

decoding the data blocks on the client computer;

integrating the data blocks together on the client computer to reconstruct an appropriate version of the original digital video; and

playing the reconstructed video on the client computer.

Google Exhibit No. 1008
Page 161

Further in accordance with a preferred embodiment of the present invention, the step of encoding includes a bit-rate control device enabling the producer to pre-select the sequence of bandwidths or quality levels for the database.

Still further in accordance with a preferred embodiment of the present invention, the step of encoding is performed in such a way that the first blocks of the database correspond to previews of the video.

Additionally in accordance with a preferred embodiment of the present invention, the steps of transmitting, decoding, integrating and playing are repeated in succession a number of times in order to transmit additional data blocks to the client, thereby upgrading the quality of the video while it is replayed.

There is also provided in accordance with another preferred embodiment of the present invention a method for encoding original digital audio data to be stored on a server computer for on-line delivery to client computers, including the steps of:

encoding the digital audio into a database including a series of encoded data blocks, each block including a sequence of encoded frames, with the property that successive blocks when decoded and integrated together provide successively higher bandwidth versions of the audio for on-line playback;

storing the database on a server computer;

processing a request by a client computer for on-line delivery of the audio in order to determine which data blocks to transmit, so as to accommodate the client bandwidth;

transmitting the necessary data blocks to the client;

decoding the data blocks on the client computer;

integrating the data blocks together on the client computer to reconstruct an appropriate version of the original digital audio; and

playing the reconstructed audio on the client computer.

Further in accordance with a preferred embodiment of the present invention, the step of encoding includes a bit-rate control device enabling the producer to pre-select the sequence of bandwidths or quality levels for the database.

Still further in accordance with a preferred embodiment of the present invention, the steps of transmitting, decoding, integrating and playing are repeated in succession a

Google Exhibit No. 1008
Page 162

number of times in order to transmit additional data blocks to the client, thereby upgrading the quality of the audio while it is replayed.

There is also provided in accordance with another preferred embodiment of the present invention, a method for encoding original digital object movie data to be stored on a server computer for on-line delivery to client including, comprising the steps of:

encoding the digital object movie into a database including a series of encoded data blocks, each block including a sequence of encoded frames, with the property that successive blocks when decoded and integrated together provide successively higher bandwidth versions of the object movie for on-line playback;

storing the database on a server computer;

processing a request by a client computer for on-line delivery of the object movie in order to determine which data blocks to transmit, so as to accommodate the client bandwidth;

transmitting the necessary data blocks to the client;

decoding the data blocks on the client computer;

integrating the data blocks together on the client computer to reconstruct an appropriate version of the original digital object movie; and

playing the reconstructed object movie on the client computer.

Further in accordance with a preferred embodiment of the present invention, the step of encoding includes a bit-rate control device enabling the producer to pre-select the sequence of bandwidths or quality levels for the database.

Still further in accordance with a preferred embodiment of the present invention, the steps of transmitting, decoding, integrating and playing are repeated in succession a number of times in order to transmit additional data blocks to the client, thereby upgrading the quality of the object movie while it is replayed.

There is also provided in accordance with another preferred embodiment of the present invention a method for encoding an original digital panorama to be stored on a server computer for on-line delivery to client computers, including the steps of:

encoding the digital panorama into a database including a series of encoded data blocks, each block including a sequence of encoded frames, with the property that

Google Exhibit No. 1008
Page 163

successive blocks when decoded and integrated together provide successively higher bandwidth versions of the panorama for on-line playback;

storing the database on a server computer;

processing a request by a client computer for on-line delivery of the panorama in order to determine which data blocks to transmit, so as to accommodate the client bandwidth;

transmitting the necessary data blocks to the client;

decoding the data blocks on the client computer;

integrating the data blocks together on the client computer to reconstruct an appropriate version of the original digital panorama; and

playing the reconstructed panorama on the client computer.

Further in accordance with a preferred embodiment of the present invention the step of encoding includes a bit-rate control device enabling the producer to pre-select the sequence of bandwidths or quality levels for the database.

Still further in accordance with a preferred embodiment of the present invention the steps of transmitting, decoding, integrating and playing are repeated in succession a number of times in order to transmit additional data blocks to the client, thereby upgrading the quality of the panorama while it is replayed.

There is also provided in accordance with another preferred embodiment of the present invention a method for encoding original digital large still image data to be stored on a server computer for on-line delivery to client computers, including the steps of:

encoding the large digital image into a database including a series of encoded data blocks, each block including a sequence of encoded multi-resolution tiles of the image, with the property that successive blocks when decoded and integrated together provide successively higher quality versions of the tiles for display;

storing the database on a server computer;

processing a request by a client computer for on-line delivery of the image in order to determine which data blocks to transmit;

transmitting the necessary data blocks to the client;

decoding the data blocks on the client computer;

Google Exhibit No. 1008
Page 164

integrating the data blocks together on the client computer to reconstruct an appropriate version of the original multi-resolution image tiles; and

interactively displaying the reconstructed tiles on the client computer.

Still further in accordance with a preferred embodiment of the present invention the step of encoding includes a compression control device enabling the producer to pre-select the sequence of quality levels for the database.

Additionally in accordance with a preferred embodiment of the present invention the step of encoding operates on a plurality of images forming an animation, and each encoded data block is comprised of multi-resolution tiles from the plurality of images.

There is also provided in accordance with another preferred embodiment of the present invention a video processing system operative on digital video data for encoding the digital video, storing it on a server computer and delivering it to client computers on-line upon request including:

an encoder for compressing the digital video into a database including a series of encoded data blocks, each block including a sequence of encoded frames, with the property that successive blocks when decoded and integrated together provide successively higher bandwidth versions of the video for on-line playback;

a storage device for archiving the database on a server computer;

a processing unit for accepting a request by a client computer for on-line delivery of the video and determining which data blocks to transmit, so as to accommodate the client bandwidth;

a transmitter for delivering the necessary data blocks to the client;

a decoder for decompressing the data blocks back into video data on the client computer;

an accumulator for integrating the data blocks together on the client computer to reconstruct an appropriate version of the original digital video; and

a player on the client computer for playing the reconstructed digital video.

Additionally in accordance with a preferred embodiment of the present invention the encoder includes a bit-rate controller enabling the user to pre-select the sequence of bandwidths or quality levels for the database.

Google Exhibit No. 1008
Page 165

Moreover in accordance with a preferred embodiment of the present invention the encoder compresses the digital video in such a way that the first blocks of the database correspond to previews of the video.

Further in accordance with a preferred embodiment of the present invention the transmitter, decoder, accumulator and player repeatedly operate in succession a number of times in order to transmit additional data blocks to the client, thereby upgrading the quality of the video while it is being replayed.

There is also provided in accordance with another preferred embodiment of the present invention an audio processing system operative on digital audio data for encoding the digital audio, storing it on a server computer and delivering it to client computers on-line upon request including:

an encoder for compressing the digital audio into a database including a series of encoded data blocks, each block including a sequence of encoded frames, with the property that successive blocks when decoded and integrated together provide successively higher bandwidth versions of the audio for on-line playback;

a storage device for archiving the database on a server computer;

a processing unit for accepting a request by a client computer for on-line delivery of the audio and determining which data blocks to transmit, so as to accommodate the client bandwidth;

a transmitter for delivering the necessary data blocks to the client;

a decoder for decompressing the data blocks back into audio data on the client computer;

an accumulator for integrating the data blocks together on the client computer to reconstruct an appropriate version of the original digital audio; and

a player on the client computer for playing the reconstructed digital audio.

Further in accordance with a preferred embodiment of the present invention the encoder includes a bit-rate controller enabling the user to pre-select the sequence of bandwidths or quality levels for the database.

Still further in accordance with a preferred embodiment of the present invention the transmitter, decoder, accumulator and player repeatedly operate in succession a

Google Exhibit No. 1008
Page 166

number of times in order to transmit additional data blocks to the client, thereby upgrading the quality of the audio while it is being replayed.

There is also provided in accordance with another preferred embodiment of the present invention an object movie processing system operative on digital object movie data for encoding the digital object movie, storing it on a server computer and delivering it to client computers on-line upon request including:

an encoder for compressing the digital object movie into a database including a series of encoded data blocks, each block comprising a sequence of encoded frames, with the property that successive blocks when decoded and integrated together provide successively higher bandwidth versions of the object movie for on-line playback;

a storage device for archiving the database on a server computer;

a processing unit for accepting a request by a client computer for on-line delivery of the object movie and determining which data blocks to transmit, so as to accommodate the client bandwidth;

a transmitter for delivering the necessary data blocks to the client;

a decoder for decompressing the data blocks back into object movie data on the client computer;

an accumulator for integrating the data blocks together on the client computer to reconstruct an appropriate version of the original digital object movie; and

a player on the client computer for playing the reconstructed digital object movie.

Further in accordance with a preferred embodiment of the present invention the encoder includes a bit-rate controller enabling the user to pre-select the sequence of bandwidths or quality levels for the database.

Still further in accordance with another preferred embodiment of the present invention the transmitter, decoder, accumulator and player repeatedly operate in succession a number of times in order to transmit additional data blocks to the client, thereby upgrading the quality of the object movie while it is being replayed.

There is also provided in accordance with another preferred embodiment of the present invention a panorama processing system operative on digital panorama data for encoding the digital panorama, storing it on a server computer and delivering it to client computers on-line upon request including:

Google Exhibit No. 1008
Page 167

an encoder for compressing the digital panorama into a database including a series of encoded data blocks, each block including a sequence of encoded frames, with the property that successive blocks when decoded and integrated together provide successively higher bandwidth versions of the panorama for on-line playback;

a storage device for archiving the database on a server computer;

a processing unit for accepting a request by a client computer for on-line delivery of the panorama and determining which data blocks to transmit, so as to accommodate the client bandwidth;

a transmitter for delivering the necessary data blocks to the client;

a decoder for decompressing the data blocks back into panorama data on the client computer;

an accumulator for integrating the data blocks together on the client computer to reconstruct an appropriate version of the original digital panorama; and

a player on the client computer for playing the reconstructed digital panorama.

Further in accordance with a preferred embodiment of the present invention the encoder includes a bit-rate controller enabling the user to pre-select the sequence of bandwidths or quality levels for the database.

Still further in accordance with a preferred embodiment of the present invention the transmitter, decoder, accumulator and player repeatedly operate in succession a number of times in order to transmit additional data blocks to the client, thereby upgrading the quality of the panorama while it is being replayed.

There is also provided in accordance with another preferred embodiment of the present invention an image processing system operative on large digital image data for encoding the digital image, storing it on a server computer and delivering it to client computers on-line upon request including:

an encoder for compressing the large digital image into a database including a series of encoded data blocks, each block including a sequence of encoded multi-resolution tiles of the image, with the property that successive blocks when decoded and integrated together provide successively higher quality versions of the image tiles;

a storage device for archiving the database on a server computer;

Google Exhibit No. 1008
Page 168

a processing unit for accepting a request by a client computer for on-line delivery of the image and determining which data blocks to transmit;

a transmitter for delivering the necessary data blocks to the client;

a decoder for decompressing the data blocks back into image tile data on the client computer;

an accumulator for integrating the data blocks together on the client computer to reconstruct an appropriate version of the original multi-resolution image tiles; and

an interactive viewer on the client computer for displaying the reconstructed image tiles.

Additionally in accordance with a preferred embodiment of the present invention the encoder includes a compression controller enabling the user to pre-select the sequence of quality levels for the database.

Moreover in accordance with a preferred embodiment the encoder operates on a plurality of images forming an animation, and each encoded data block is comprised of multi-resolution tiles from the plurality of images.

There is also provided in accordance with another preferred embodiment of the present invention a method for caching of data which gets transmitted from servers to clients on a central hub within a network, including the steps of:

encoding digital multimedia data into databases including a series of encoded data blocks, each block including a sequence of encoded frames, with the property that successive blocks when decoded and integrated together provide successively higher bandwidth versions of the media for on-line playback;

storing the databases on a multitude of server computers;

managing within the hub requests by client computers for on-line delivery of media stored on server computers in order to determine which data blocks to transmit, so as to accommodate the client bandwidth;

transmitting the necessary data blocks from the server and from the hub to the client;

storing the data blocks delivered by the server in the cache residing in the central hub;

processing within the hub the data blocks it receives;

Google Exhibit No. 1008
Page 169

decoding the data blocks on the client computer;

integrating the data blocks together on the client computer to reconstruct an appropriate version of the original digital media; and

playing the reconstructed media on the client computer.

Further in accordance with a preferred embodiment of the present invention the step of managing is performed by:

setting inventory flags to indicate which data blocks are currently stored in the hub.

Still further in accordance with a preferred embodiment of the present invention the step of managing further including the steps of:

communicating with the servers to monitor which media data is outdated;

removing from cache the blocks corresponding to the media data which is outdated; and

resetting the inventory flags to indicate that the above blocks are no longer stored in the cache.

Still further in accordance with a preferred embodiment of the present invention the step of processing comprising the steps of:

decoding the data blocks received;

integrating the data blocks together to reconstruct appropriate versions of the original digital media; and

encoding the reconstructed media versions into an intermediate database for future transmission to the clients.

There is also provided in accordance with another preferred embodiment of the present invention a proxy system operative on a server/client network for caching of data which gets transmitted from servers to clients on a central hub, including:

an encoder for compressing digital multimedia data into databases including a series of encoded data blocks, each block including a sequence of encoded frames, with the property that successive blocks when decoded and integrated together provide successively higher bandwidth versions of the media for on-line playback;

server communication lines from the servers to the hub for sending data blocks;

client communication lines from the hub to the clients for sending digital data;

storage devices for archiving the databases on a multitude of server computers;

Google Exhibit No. 1008
Page 170

a management unit within the hub to process requests by client computers for on-line delivery of media stored on server computers in order to determine which data blocks to transmit, so as to accommodate the client bandwidth;

a transmitter for delivering the necessary data blocks on the server communication lines from the server to the hub, and on the client communication lines from the hub to the client;

a storage device for saving the data blocks delivered by the server communication lines in the cache residing in the central hub;

a processing unit within the hub for processing the data blocks which the hub receives;

a decoder for decompressing the data blocks on the client computer;

an accumulator for integrating the data blocks together on the client computer to reconstruct an appropriate version of the original digital media; and

a player for playing the reconstructed media on the client computer.

Further in accordance with a preferred embodiment of the present invention the management unit operates by setting inventory flags to indicate which data blocks are currently stored in the hub.

Still further in accordance with a preferred embodiment of the present invention the management unit operates by monitoring from the servers which media data is outdated, removing from cache the blocks corresponding to the media data which is outdated, and resetting the inventory flags to indicate that the above blocks are no longer stored in the cache.

Additionally in accordance with a preferred embodiment of the present invention the processing unit comprising:

a decoder for decompressing the data blocks received;

an accumulator for integrating the data blocks together to reconstruct appropriate versions of the original digital media; and

an encoder for compressing the reconstructed media versions into an intermediate database for future transmission to the clients.

Google Exhibit No. 1008
Page 171

There is also provided in accordance with another preferred embodiment of the present invention a multi-casting unit (MCU) system operative on a broadcasting network for caching of data which gets transmitted from stations to viewers, including:

an encoder for compressing digital multimedia data into databases including a series of encoded data blocks, each block including a sequence of encoded frames, with the property that successive blocks when decoded and integrated together provide successively higher bandwidth versions of the media for on-line playback;

station communication lines from the stations to the MCU for sending data blocks;

viewer communication lines from the MCU to the viewers for sending data;

viewer receiver units for receiving the data sent by the MCU;

storage devices for archiving the databases on a multitude of station computers;

a management unit within the MCU to process requests by viewers for on-line delivery of media stored on station computers in order to determine which data blocks to transmit, so as to accommodate the viewer bandwidth;

a transmitter for delivering the necessary data blocks on the station communication lines from the station to the MCU, and on the viewer communication lines from the MCU to the viewer receiver units;

a storage device for saving the data blocks delivered by the station communication lines in the cache residing in the MCU;

a processing unit within the MCU for processing the data blocks which the MCU receives;

a decoder for decompressing the data blocks on the viewer receiver;

an accumulator for integrating the data blocks together on the viewer receiver unit to reconstruct an appropriate version of the original digital media; and

a player for playing the reconstructed media from the viewer receiver unit.

Further in accordance with a preferred embodiment of the present invention the management unit operates by setting inventory flags to indicate which data blocks are currently stored in the MCU.

Still further in accordance with a preferred embodiment of the present invention the management unit operates by monitoring from the stations which media data is outdated, removing from cache the blocks corresponding to the media data which is

Google Exhibit No. 1008
Page 172

outdated, and resetting the inventory flags to indicate that the above blocks are no longer stored in the cache.

Additionally in accordance with a preferred embodiment of the present invention the processing unit within the MCU including:

a decoder for decompressing the data blocks received;

an accumulator for integrating the data blocks together to reconstruct appropriate versions of the original digital media; and

an encoder for compressing the reconstructed media versions into an intermediate database for future transmission to the viewers.

There is also provided in accordance with another preferred embodiment of the present invention a method for streaming multimedia data over a network, including the steps of:

encoding the media into a progressive database indexed according to frame and progressive block numbers;

serializing the encoded database;

storing the serialized database on a server;

streaming the serialized database to a client upon request;

creating a mirror copy of the encoded database on the client computer from the data which streams in; and

decoding the encoded database on the client computer into a sequence of frames for real time display.

There is also provided in accordance with another preferred embodiment of the present invention a multimedia network streaming system, including:

an encoder for compressing the media into a progressive database indexed according to frame and progressive block numbers;

a sequencer for serializing the encoded database;

a storage device for archiving the serialized database on a server;

a transmitter for streaming the serialized database to a client upon request;

a processor for creating a mirror copy of the encoded database on the client computer from the data which streams in; and

Google Exhibit No. 1008
Page 173

a decoder for decompressing the encoded database on the client computer into a sequence of frames for real time display.

There is also provided in accordance with another preferred embodiment of the present invention a system for transmitting model based data representations of three dimensional images over plural transmission links having limited bandwidth, said system including:

a digital data source storing model based data representations of three dimensional images;

an image processor for rendering views of said model based data representations into raster bitmap format;

a digital data receiver receiving said digital data in said raster bitmap format over a one of the plural transmission links having limited bandwidth; and

a digital data transmitter operative to transmit the digital data in said raster bitmap format to said receiver over a transmission link having a limited bandwidth in plural blocks which are sequentially transmitted at a rate determined by the limited bandwidth, each block being an incomplete collection of data which includes parts of multiple frames, each frame being viewable in a selectable order by said receiver even when less than all of the plural blocks have been received, receipt of subsequent blocks by the receiver being used to cumulatively improve the quality of the digital data viewed by the receiver.

Further in accordance with a preferred embodiment of the present invention the model based data representations comprise VRML representations.

Still further in accordance with a preferred embodiment of the present invention the model based data representations comprise CAD-CAM representations.

Additionally in accordance with a preferred embodiment of the present invention the image processor is operative to render only views which are selected by a user.

There is also provided in accordance with another preferred embodiment of the present invention a method for transmitting model based data representations of three dimensional images over plural transmission links having limited bandwidth, said system including:

storing model based data representations of three dimensional images;

Google Exhibit No. 1008
Page 174

rendering views of the model based data representations into raster bitmap format;

receiving the digital data in said raster bitmap format over a one of said plural transmission links having limited bandwidth; and

transmit the digital data in said raster bitmap format to the receiver over a transmission link having a limited bandwidth in plural blocks which are sequentially transmitted at a rate determined by the limited bandwidth, each block being an incomplete collection of data which includes parts of multiple frames, each frame being viewable in a selectable order by the receiver even when less than all of the plural blocks have been received, receipt of subsequent blocks by the receiver being used to cumulatively improve the quality of the digital data viewed by the receiver.

Further in accordance with a preferred embodiment of the present invention the model based data representations comprise VRML representations, and CAD-CAM representations.

Still further in accordance with a preferred embodiment of the present invention the image processor is operative to render only views which are selected by a user.

The following definitions are employed throughout the specification and claims:

RESOLUTION--The relationship between the number of digital samples per unit of an original and the number of digital samples per unit in a rendered version thereof. Specifically, when dealing with images, resolution refers to the relationship between the number of pixels per unit area of an original image or scene and the number of pixels per unit area in a displayed image. Specifically, when dealing with audio, resolution refers to the relationship between the number of samples per unit time of an original sound and the number of samples per unit time in a played sound.

QUALITY--The degree to which a rendered version of an original is faithful to the original. Specifically, when dealing with images, quality refers to the degree to which the displayed image is faithful to the original image or scene. Normally this is expressed as the degree to which the approximation of pixel values in the displayed image approaches the correct pixel values in the original image or scene. Specifically, when dealing with audio, quality refers to the degree to which a played sound is faithful to the original sound.

Google Exhibit No. 1008
Page 175

FRAME--A portion of an original which can be independently and interactively manipulated. Specifically, when dealing with images, frame refers to a portion of an image or of a collection of images which can be independently and interactively manipulated. Specifically, when dealing with audio, frame refers to a portion of a sound which is delimited in time and can be independently and interactively manipulated.

BLOCK--A sequentially transmitted collection of partial data which is used to build multiple frames. The frames are built up of one or more sequentially transmitted blocks, whose contents are accumulated. Specifically, when dealing with images, the block contains image data. Specifically when dealing with audio, the block contains audio data.

PARTIAL FRAME--The part of a frame which is contained in a given block.

TILE--A window sized pixel array of a predetermined given size forming part of an image. For example, tiles partition an image into a plurality of arrays, each of which contains an identical number of pixels.

Google Exhibit No. 1008
Page 176

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be understood and appreciated more fully from the following detailed description, taken in conjunction with the drawings in which:

Fig. 1 and Fig. 2 are simplified block diagrams illustrating a system for scalable representation of multimedia data for progressive asynchronous transmission, constructed and operative in accordance with a preferred embodiment of the present invention;

Fig. 3A is a simplified schematic diagram of the database structure of the present invention which includes three databases, embodied within the client-server system of the present invention;

Fig. 3B is a simplified diagram of a database structure particularly useful in the client database of Fig. 3A, illustrating its two-dimensional nature;

Fig. 4 is an illustration of the operation of a preferred embodiment of the present invention;

Fig. 5 is a simplified schematic diagram of a production tool for converting a digital multimedia file into a progressive scalable database representation for storage on a server computer in accordance with a preferred embodiment of the present invention;

Fig. 6 is a simplified schematic diagram of the structure of a block within the server database, partitioned into frames which can be accessed randomly in accordance with a preferred embodiment of the present invention;

Fig. 7 is a simplified schematic diagram of a decoder for receiving and integrating data blocks from a scalable database, to form a version of a digital multimedia object for playback in accordance with a preferred embodiment of the present invention;

Fig. 8 is a simplified schematic diagram of a scalable progressive database for a video clip in which the first data blocks are used for previewing the video in accordance with a preferred embodiment of the present invention;

Fig. 9 and Fig. 10 are simplified schematic diagrams of a system for incorporating a scalable progressive database into a time-based video sequence of frames indexed by two time scales: a macro and micro scale, in accordance with a preferred embodiment of the present invention;

Google Exhibit No. 1008
Page 177

Fig. 11 is a simplified block diagram of a proxy system used to cache in a central hub multimedia data which is transmitted from servers to clients in accordance with a preferred embodiment of the present invention;

Fig. 12 is a simplified block diagram of a system for generating a scalable database from digital media data, by running a compressor in a feedback loop in accordance with a preferred embodiment of the present invention;

Fig. 13 is a simplified block diagram of a decoder for the database generated by the system of Fig. 12 in accordance with a preferred embodiment of the present invention;

Fig. 14 is a simplified diagram illustrating a scalable progressive database useful for a large still image in accordance with a preferred embodiment of the present invention;

Fig. 15 is a simplified diagram illustrating a virtual reality system constructed and operative in accordance with a preferred embodiment of the present invention;

Fig. 16 is a simplified flowchart illustrating operation of the system of Fig. 15; and

Fig. 17 is an illustration of the operation of a preferred embodiment of the present invention illustrated in Figs. 15 and 16, permission to reproduce Fig. 17 was granted by Tecnomatix, Ltd.

Google Exhibit No. 1008
Page 178

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The preferred embodiments of the present invention may be better understood given the following technical and theoretical explanation.

The present invention provides a novel method for representing multimedia data. The invention provides a scalable representation, so that the data can be asynchronously transmitted to clients having different bandwidth connections, played on-line almost immediately after the transmission begins, interactively controlled, and also progressively upgraded as it is replayed.

Although the present invention is described hereinbelow with particular reference to image data, it is to be appreciated that it is applicable also to non-image data, such as audio data.

When addressing bandwidth limitations, it is natural to think in terms of data rates, or velocities. Video players, for example, play at standard rates such as thirty frames per second (fps), and require the images for display to be available at this rate. If the images are already stored on a local hard disk, then all that is necessary is disk access, which is very fast. On the other hand, if the images are streamed in from a server, then in order for on-line playback to be possible before a full download is finished, the rate of transmission must be great enough to supply the frames at thirty fps. This does not mean that the network link has to transmit the data equivalent to thirty full frames every second. Due to compression, it suffices if the network transmits thirty *compressed* frames every second.

For example, if the compression achieved is 10:1, then it suffices to transmit at a rate of three fps, provided that the client CPU can decompress thirty compressed frames into full frames every second. In fact, compression is the mediator between the video player and the bandwidth. The player does not slow down when bandwidth is low; rather, the compression ratio has to be greater. Should a bottleneck arise, and a frame is not available when the player needs it, then the player simply skips that frame, but continues to expect frames at the thirty fps rate. The video can be preset at the outset for lower rates than thirty fps, but not much lower, since slow video playback breaks the continuity between frames, and thus loses the effect of motion.

Google Exhibit No. 1008
Page 179

Thus, realizing that higher compression means lower quality, it can be appreciated that users having high bandwidth connections can receive high quality video for on-line playback, and users having low bandwidth connections receive low quality video. In a non-scalable setting, it is thus necessary to prepare different compressed media files for each playback rate and bandwidth combination. For example, four different versions of the media could exist for (a) 24 fps playback, 14.4 Kbs bandwidth, (b) 24 fps playback, 28.8 Kbs bandwidth, (c) 30 fps playback, 14.4 Kbs bandwidth, (d) 30 fps, 28.8 Kbs bandwidth. As described below, using the present invention a single media file can be used to accommodate all four of these combinations.

In the present invention, the media data is comprised of m frames $F_1, F_2, \ldots, F_m$. A frame can be, for example, an individual frame of a movie sequence, a piece of a panoramic view, an individual segment of an audio signal, or even a sub-sampled version of a large still image. It can also be a group of such frames, such as for example, a group of inter-frames between key frames in a video segment, in a case where an H.263 codec is being used. In broad terms, frames are units of interactivity. For example, in object movies where interactivity means frame advance, a frame unit is an individual still image, whereas in gazing applications where interactivity means zooming in and out, the frame units are multi-resolution tiles.

The representation encodes the media data into n data blocks $B_1, B_2, \ldots, B_n$ preferably of roughly equal size. Each encoded data block B_j contains m compressed frame units $F_1^j, F_2^j, \ldots, F_m^j$. Thus it is appreciated that the database is arranged in two dimensions, corresponding to blocks and frames. The dimension used for blocks is for achieving progressiveness, and the dimension used for frames is for achieving interactivity. The frame data can be transmitted in a selective order, but the blocks must be transmitted in sequence, since they build cumulatively. This is an essential feature of the subject invention.

Data block B_1 is used to deliver the media at the lowest bandwidth, say f_1 Kbs; data blocks B_1 and B_2 when integrated together, are used to deliver the media at bandwidth $f_2 = 2f_1$ Kbs; and in general, for $1 \leq k \leq n$, data blocks $B_1, B_2, \ldots, B_m$ when integrated together, are used to deliver the media at bandwidth $f_k = kf_1$ Kbs. Each higher bandwidth version delivers a higher quality rendition of the media. In this way, the representation

Google Exhibit No. 1008
Page 180

can accommodate multiple clients connecting to the network with different bandwidths. Moreover, a client connecting with bandwidth f_1 who has downloaded data block B_1 and played the media, can continue downloading block B_2 in background, since the bandwidth has been freed, and thereby achieve the same quality as bandwidth f_2 the next time the media is replayed.

Within each block the frames can be accessed randomly and delivered selectively, so that the user can vary the quality level among the frames. For example, a viewer who wants to gaze at frame #3 may instruct the database to send frame #3 data from the first ten blocks, but only one block of data for all of the other frames. The viewer selection is carried out interactively, through the use of keyboard presses and mouse clicks, as the media is being played. Whereas for some applications it may be most natural to transmit the entire blocks in sequence, for other applications it may be more effective to first deliver as much data as possible for specific frames at the expense of lowering the quality of other frames. The two dimensionality of the database, together with its interactive accessibility, gives the user complete control over the transmission sequence.

The scalable representation that is the subject of the present invention is embodied in a production tool which enables the producer to control the bandwidth parameters f_k, or equivalently, the qualities of the media versions obtained by integrating blocks B_1, B_2, ..., B_k. In general, it is not necessary that the blocks be of equal size, nor that the frequencies f_k be given by kf_1, although this is the preferred embodiment. The production tool also enables the producer to control the final quality of the highest bandwidth version, or equivalently, the total number, n, of data blocks in the representation.

In contrast, if the media data representation is not scalable, but is encoded instead for a specific bandwidth f, then only clients with bandwidth f or greater can play the media on-line as it is being downloaded. A client with a lower bandwidth than f would have to download the entire data stream to memory in order to begin playback, which can take a great deal of time on account of the large file sizes typically used in multimedia production. A client with a higher bandwidth connection than f would not be able to take advantage of it to receive higher quality media. Moreover, there would be no means of upgrading media quality, even for clients with high bandwidth connections,

Google Exhibit No. 1008
Page 181

other than to transmit an entirely new data stream from the server side, and discard the previously downloaded data.

Applications of the invention include, inter alia, scalable audio and video transmission, video previewing, progressively rendered object movies and panoramas, large still images, efficient proxy or multi-casting unit (MCU) management for web and other hubs, and VRML transmission, as described hereinafter in greater detail.

Scaleable audio transmission: Digital audio data can be progressively encoded into a scalable database for asynchronous transmission at different bandwidths. A client connected with a low bandwidth line can receive a low quality version of the audio, which can be played back on-line at the low bandwidth as the data streams in. After the audio is played, additional data blocks can continue to be received and integrated with the previous blocks, so that the audio is upgraded to higher quality for replay.

Scaleable video transmission: Similar to the description above for the audio transmission, digital video data can be encoded into a scalable database for asynchronous delivery and progressive quality upgrade.

For certain time-based video systems, there is disclosed a novel way to incorporate the scalable database so that progressiveness and immediate playback can be achieved, even in a single-play mode. Specifically, this applies to video systems with two time scales, such as is present in the Apple QUICKTIME® movie player. The first time scale (hereinafter referred to as the "major scale") is used to advance from one frame to the next, based on major units of time. The second time scale (hereinafter referred to as the "minor scale") is a sub-division of the major scale into smaller time units, and is used to incorporate small changes or fluctuations into the frame being displayed. For example, the major scale can be advancing through a movie of a bird flying and the minor scale can be adding fluttering to the bird's wings. The advantage of such a two-scale player is that the decoder, which does the intensive processing to supply the frames, need only run at the slower rate, e.g. 3 frames per second (fps), governed by the major scale, whereas the viewer, doing the less intensive processing, is playing at the faster rate, e.g. 30 fps, governed by the minor scale.

The subject invention can be incorporated into a system having two time scales as described hereinabove by using the minor scale in a way different from the way that was

Google Exhibit No. 1008
Page 182

originally intended. Instead of being used to introduce fluctuations, it is used to display progressively rendered versions of a frame. At each minor time unit, the player displays the latest version available of the frame indexed by the major scale. For example, suppose there are ten minor time units within the major time unit during which frame #4 is to be displayed. At the first minor time unit, the player initially displays the version of frame #4 which it has available from the first data blocks already processed. As additional blocks of data are accumulated and higher quality versions of frame #4 become available, the player displays those frames at successive minor time units. This continues for ten minor time units, until the next major time unit, at which time frame #5 is to be displayed. The cycle then repeats, and the version of frame #5 which is already available is initially displayed at the first minor time unit. Thus it can be seen that progressiveness can be achieved by interjecting into the minor scale, the versions of the frames obtained by accumulating successive blocks.

For the user to be able to view the video immediately, without waiting for the entire file to download, the production tool must store the encoded video in the order of successive blocks. Each partial frame must be handled as if it were an entire frame. That is, the production tool must treat the movie as if there were a total of $m \cdot n$ distinct frames being encoded. On the other hand, each frame is sent only once to the codec for encoding, and is returned as a series of encoded partial frames. Thus it is necessary to post-process the encoded data file, to rearrange the data items from a frame dominated order to a block dominated order. This rearrangement process is referred to as "flattening" in the art.

The player in turn, however, must know that although it is receiving what appears to be $m \cdot n$ data items, there are really only a total of m frames. It must decode and accumulate every successive sequence of m data items with the previous ones, to update the frames. The combined effect of the flattening on the production side and the player's interpretation on the client side enables seamless integration of the scalable progressive database within a non-progressive video interface. That is, the incorporation of progressive blocks does not require any modifications to the existing interface.

Video previewing: When encoding digital video data into a scalable database, the first data blocks can be used to generate a preview of the video, restricted to selected

Google Exhibit No. 1008
Page 183

frames. The preview can be played back by the client almost immediately after the streaming begins. Moreover, additional data blocks received are integrated with the data blocks from the preview, to form full view versions of the video.

Object movies: Advertising agencies are using object movies to produce interactive 3-D virtual reality presentations of merchandise on the Internet. The user can rotate and zoom the 3-D object, and examine it from different viewing angles. Using the methodology of the current invention, object movies can be progressively encoded so that the viewer can download and begin playing them almost immediately after the streaming begins. Initially the movie will scale to a quality commensurate with the bandwidth of the user's network connection, but as the data blocks are received and the user interacts with the movie, additional data blocks are delivered and integrated with the previous blocks, resulting in a higher and higher quality movie. An important feature of the invention is that, regardless of bandwidth, the user can begin playback and interaction almost immediately, and does not need to wait for the complete download, as the first version of the movie delivered scales itself to the native bandwidth. As playback continues additional data streams in the background and the movie version is upgraded to higher and higher quality.

Panoramas: Panoramas are very large images which the user cannot view in their entirety, but rather sees within a restricted viewing window. By panning in various directions, and zooming in and out, the user navigates through the panorama. The continuous change in viewing window gives the effect of movement within a scene. Similar to the description above for object movies, panoramas can be progressively encoded so that the viewer can download and begin navigating through them almost immediately after the streaming begins. Initially the panorama scales to the client bandwidth, and after the first data blocks are received, additional data blocks are streamed in background while the panorama is playing, to provide higher and higher image quality.

Large still images: Although large high quality still images are not composed of frames in the conventional sense, the large sizes of the image data files makes the present invention an effective means for interactive on-line viewing. The frames can be small image tiles within the full image at different resolutions, the smaller tiles having higher

Google Exhibit No. 1008
Page 184

resolution than the larger ones. The locations of the tiles can be marked as hot spots. When the viewer clicks on a hot spot within a specific tile, the database delivers that tile at a higher resolution, giving the effect of a zoom in. Within the higher resolution tile there can be more hot spots, and the zooming can continue through the database.

As a simple example, the first frame within a data block may contain the full image sub-sampled by 4:1, for example, in each dimension. The next set of frames within the data block may contain (some subset of) the four quadrants of the full image sub-sampled by 2:1 in each dimension. The next set of frames within the data block may contain (some subset of) the sixteen quadrants within the above four quadrants at the original resolution. A viewer could see the 4:1 reduction of the original image (the first block), click on one of the quadrants and then see that quadrant at a 2:1 reduction (a frame from the second set), and click further on one of its quadrants and then see it at full scale (a frame from the third set).

Efficient proxy/MCU management: Proxies are large storage devices, located as hubs within networks, used as large caches for data being delivered from servers to clients. Similarly MCUs are large storage devices used as caches for data being delivered from broadcasting stations to viewers, such as cable TV. As data is streamed from servers to clients or from broadcasting stations to viewers upon request, the proxy or MCU stores the data in a central hub so that it is available for delivery at a high bandwidth if requested again by any of the clients connected to the hub. It plays a similar role to paging files on a local computer disk, but on a much larger scale and for a much larger clientele.

The scalable representation of the subject invention is particularly well suited for proxies and MCUs which operate in asynchronous environments. Server/client connections and broadcast transmissions can be of many different bandwidths, and so the proxy or MCU can be accumulating versions of the same multimedia data corresponding to different qualities. Without scalability these versions are all independent of one another, and cannot be combined to achieve quality levels other than those originally preset or combined to save space. Using the scalable representation of the subject invention, the proxy or MCU can be optimized to cache the various progressive building blocks. This affords great flexibility in being able to create versions of different

Google Exhibit No. 1008
Page 185

quality levels, and reduces the space requirements.

As a simple example, a first user with a low bandwidth connection f_1 to the server, who demands a multimedia file, downloads data block B_1, which is then also saved to cache on the proxy, and the user receives a low quality version (quality level 1) of the media. A second user with a higher bandwidth connection f_2 to the server, who demands the same multimedia file, can download data blocks B_2 and B_3 from the server, and can access block B_1 directly from the proxy. The three data blocks are integrated and the second user receives a very high quality version (quality level 3) of the media. Data blocks B_2 and B_3 would then also be stored on the proxy. A third user with a direct connection to the proxy of bandwidth f_2 who now demands the same multimedia file, can receive the high quality version (quality level 2) comprising of blocks B_1 and B_2 directly from the proxy. Without a scalable representation it would not be possible for the second user, with a bandwidth connection of f_2, to receive a quality level 3 version corresponding to the higher bandwidth f_3, nor for the proxy to deliver to the third user a version at a quality level different from those directly available in its cache. Moreover, without a scalable representation the proxy memory required to cache the f_1, f_2 and f_3 versions would equal the size of six data blocks, rather than three. It can thus be seen that the proxy or MCU inherits the scalability from the servers, giving it a great deal more flexibility in its media delivery to the clients than would be possible in a non-scalable environment.

VRML transmission: Virtual reality modeling language (VRML) is a descriptive language for representing and rendering three-dimensional objects. The objects are modeled as collections of polygonal elements, the description of which forms a VRML database. Depending on the viewing parameters, the individual elements are processed and the desired view of the object is rendered into a raster bitmap for display. The VRML representation is rich enough to encapsulate all possible views of the object. In fact, there is an infinity of possible variations in viewing parameters. A user interacts with the VRML object by adjusting viewing parameters, through mouse clicks and keyboard presses. VRML was first popularized by Silicon Graphics. Their top Iris workstations, for example, can render on the order of a million polygonal elements per second. VRML images are characterized by their sharp photo-realistic attributes.

Google Exhibit No. 1008
Page 186

The present invention can be applied to efficiently deliver VRML imagery over a server/client network, for on-line interaction. Within the VRML environment itself, two problems arise when transmitting VRML databases over a network for on-line interaction. First, there is the bandwidth limitation, which inhibits the rate of transmission. Second, there is the intense processing on the client side, necessary to render the many polygonal elements into a bitmap for viewing. It would be preferable to have the server (typically a more powerful computer) do the rendering, but then it would be necessary to store rendered bitmaps of every possible set of viewing conditions – a feat that would require on the order of terabytes of disk space.

The present invention can be used to mitigate the problem by allowing the rendering to be done on the server computer without requiring enormous memory, and yet enable the client to freely interact with the VRML object in an on-line interactive setting. This is one of many examples involving real-time encoding. The invention operates by receiving the viewing parameters from the user, rendering the corresponding image on the server into a raster bitmap image, encoding the bitmap into progressive partial frames and inserting them into a two-dimensional server database. The encoded data within the server database is continually streamed from server to client, enabling the client to begin viewing a low quality image as soon as the first partial frame data arrives. As the user navigates through the VRML, additional bitmaps are rendered, encoded and inserted into the server database. Whenever the user re-traces steps, so that the viewing parameters are the same as those selected at some previous stage, the server does not need to render the same bitmaps again. Rather, the streaming simply continues in background, and the quality of the image on the client side is enhanced as additional partial frames are integrated. Similarly, if the user stays focused on a single view, then the bitmap being displayed is enhanced as additional partial frames stream in. Once all of the partial frames are integrated, the image has the same sharp photo-realistic quality as is characteristic of VRML images. On the other hand, the user does not have to wait for all of the data to arrive in order to interact with the object, nor does the client computer have to do the intensive processing to render the VRML database into bitmaps.

Reference is now made to Fig. 1 which shows a block diagram of a system for providing on-line virtual reality (VR) movies. The system includes a production

Google Exhibit No. 1008
Page 187

workstation 3 for receiving input images and processing same, as will be described hereinafter. Such input images may be constituted by photographs which are scanned into the workstation. The output from production workstation 3, being a raw VR movie, is fed into an encoder 5 for preparing the movie for transmission and in turn, applied to a server 7 essentially used for storage and transmission of the movie to clients, namely, subscribers or user units 9.

Reference is now made to Fig. 2, which shows user unit 9 of Fig. 1 in greater detail. Seen is a transceiver 34, an asynchronous memory/database 35, a decoder 36 and a user's workstation 40.

Typical operation of a preferred embodiment of the present invention is now described with reference to Figs. 1 and 2. Selected images are introduced in production workstation 3 in which the VR movie is produced in accordance with a certain script. The producer at the workstation determines the number and size (for example, in bytes) of the partial frames and also defines the various available interactions between the frames by defining hot spots and objects at 21, using auxiliary standard devices for producing movies, such as a keyboard, a mouse, speakers and a CPU all designated by 23. The product obtained is a raw VR movie, which is a complete VR movie that has not been reformatted for transmission. The preparation of the movie for transmission is effected in encoder 5 where partial frames are generated through an iterative process.

The partial frames are generated by encoder 5 as controlled by the controller 25, as follows A partial resolution frame or a partial resolution slice of each frame of the VR movie sequence is generated. One example of such a partial resolution frame is sub-sampled scan lines, e.g., the removal of every 10th line of a 150 line frame or a compression encoded frame, which partial resolution results in a blurry display. The partial resolution frame is then subtracted from the original frame by a partial frame subtractor 27, yielding a residual frame or a remainder frame. This process can be repeated on the residual frame, generating a second partial resolution frame. The procedure is also repeated time and time again, until the number (which is determined by the producer) of partial resolution frames is generated. The net result is a set of partial resolution frames that can be recombined into the original full resolution frame.

When this process is completed on each frame of the sequence, the partial

Google Exhibit No. 1008
Page 188

resolution frames are transmitted. The order of transmission follows the script given by the producer, and commonly, the first partial resolution frame of each frame is transmitted, followed by the second partial resolution frame, and so on. This sequence of transmission allows for the whole sequence to be viewed in a partial resolution format that progressively comes into focus.

Partial resolution frames may be optionally compression encoded, possibly taking into account similarities between various frames. This is effected in the compressor 29.

Encoded and compressed VR movie parts are passed to a server 7 where the movie parts are stored in a database 31 and transmitted to a user's unit 9 part by part, by means of a transceiver 33.

As seen in Fig. 2, a user's transceiver 34 receives movie parts and transmits requests for additional information. A user's database 35 is progressively updated with requested images or, alternatively, may be progressively updated by the server 7. User's database 35 functions asynchronously, supplying the frames to the user via a decoder 36 by request independent of data transmission. Upon receiving the frames, the decoder initially decompresses the frames as indicated at 37 (if compression took place) and then decodes and recombines them by means of a partial frame integrator 38. Following this, the partial frames are stored in the user's database where the frames may be stored in a compressed format, effected by a compressor 39. A user's workstation 40 enables the user to view and interact with the VR movie. The user utilizes the workstation for sending requests for images to the decoder which retrieves (and decompresses, if necessary), the images from the user's database and sends requests for particular images which may not yet have been transmitted to the server database. Furthermore, the user's workstation actuates any script produced in the production workstation 3. Hence the user's workstation also includes the standard devices included in the workstation 3 and designated by the number 23 (Fig. 1).

As a preface to the explanation of the remaining figures, it is important to understand that the progressive scalable database which is a subject of the current invention preferably is stored in three databases within the server/client system.

Reference is now made to Fig. 3A which shows a preferred database structure in accordance with a preferred embodiment of the present invention. It is a particular

Google Exhibit No. 1008
Page 189

feature of the present invention that three databases are employed, a server database 41, which is arranged in a serial form, containing multiple data blocks, each including multiple partial frame data, a client database 42 which is arranged in a two dimensional structure, conceptually illustrated in Fig. 3B and an interactive database 43 which contains a single data block including multiple frames, which is dynamically updated from the client database 42. In an alternative embodiment of the invention, wherein extremely high processing speeds are available at the client, the client database could be eliminated.

It is a particular feature of the present invention that interactive data streaming is provided. The use of three databases as described above enables interactive data streaming to be achieved in an efficient and cost effective manner. The use of databases having a two-dimensional structure greatly simplifies the data processing.

Server database 41, which is archived on a server constitutes a first database of the progressive scalable database. The server database 41 includes a plurality of data blocks in encoded form. As seen particularly in Fig. 3B, the progressive scalable database is two-dimensional in nature. It has a progressive dimension indexed by block number, and an interactive dimension indexed by frame number but it is serialized for streaming and can only be accessed sequentially. Server database 41 is streamed from server to client via the transmission and buffering protocol of the Internet browser.

Client database 42, the second database, is built up on the client side as the information streams in, to mirror the server database 41. Client database 42 is truly two-dimensional, with random access capability within the data blocks. The data blocks within it are also in encoded form.

Interactive database 43, the third database, is created by decoding the data from the client database 42. This interactive database 43 is one-dimensional, and contains only one sequence of frames, but it is dynamically updated. As additional block data is integrated, these frames are updated, with the previous versions over-written.

When a frame has been updated, the encoded frame used to update it is deleted from client database 42. Thus while interactive database 43 is being created and updated, client database 42 is being deleted. Since the updating of the frames is asynchronous, however, client database 42 typically contains frames from many different blocks simultaneously at any point in time. In essence, then, the progressive dimension

Google Exhibit No. 1008
Page 190

of the database is being realized through a time dimension in interactive database 43.

Interactive database 43 is controlled by the user interface through keyboard presses and mouse clicks. The creation and update of interactive database 43 from client database 42 is done in background time slices, while the client CPU is idle. Interactive database 43 may store the frames in either raw bitmap form or in an intermediate compressed form, as long as the intermediate compression is such that the frames can be decompressed in real time for display. An advantage of using an intermediate compression is to confine interactive database 43 to internal RAM, which has fast access time, rather than swap to hard disk memory, which has slow access time. The swapping in itself is a drain on processing speed.

When the user requests a frame to be displayed, interactive database 43 displays that frame immediately, if it is available. In case the frame is not available, interactive database 43 passes a message back to client database 42 requesting that frame. Client database 42 accesses the specific frame requested from its first encoded data block, if it is available, and sends it to the decoder for decompression and integration, and subsequent incorporation into interactive database 43. Once a frame is incorporated, interactive database 43 displays the frame at once. If client database 42 has not yet received the requested frame from the server stream, then it must wait until the encoded frame arrives, since the streaming is sequential. If the streaming were instead random access, client database 42 would be able to directly request the specific frame it needs from server database 41.

In summary, the server database is two-dimensional but serialized for sequential streaming; the client database is two-dimensional with random access within blocks; and the interactive database is one-dimensional but dynamically updated. In the interactive database, the progressive database dimension is actually being represented as time rather than space. This "three database strategy," using three different databases: (*i*) two-dimensional serialized, (*ii*) two-dimensional, (*iii*) one space and one time dimension, is a key to the present invention, and to the discussion of the figures in detail which follow.

For each of these three databases, the progressive dimension manifests itself in a different way. Within the server database, *progressiveness means quality*. The encoder builds the blocks of the database based on achieving the best quality at given bit rates.

Google Exhibit No. 1008
Page 191

Within the client database, *progressiveness means cumulative integration*. Progressiveness within the client database is a computational property. The accumulator on the client computer integrates frames from successive blocks with those from previous blocks. Finally, within the interactive database, *progressiveness means time*. As time progresses, the frames are dynamically updated as more blocks have been accumulated. The transmission from the server database to the client database is streamed serially, and this is where the progressive dimension is effectively converted from "space" to "time."

The progressiveness manifests itself in bandwidth during the streaming. The transmission from the client database to the interactive database is asynchronous. The client database is being created in the background while the interactive database is being played, and the former acts as a buffer for the latter. Moreover, the client can interact with the media almost immediately after the streaming begins, and does not have to wait for the client database to be constructed entirely.

The interactive dimension of the database corresponds to whatever functionality the user interface allows. For example, it can manifest itself as frame advance for videos and object movies, navigating for panoramas, and gazing for large still images.

Reference is now made to Fig. 4 which illustrates one application of the three database structure described hereinabove in Figs. 3A and 3B. For simplicity, referring additionally to Fig. 3B, each image in Fig. 4 is built up of corresponding partial frames in successive data blocks which are cumulatively received. Thus, for simplicity, one may consider the five images in a first horizontal row, to correspond to five interactively viewable frames in a first block of data, each successive frame typically illustrating a successive position of an imaged model.

Considering each successive horizontal row in Fig. 4, it can be seen that the quality of the images improves successively from the top row to the bottom row. This corresponds to the quality improvement sensed by a viewer as successive blocks of data are incorporated in the information made available to the viewer.

It is appreciated that in accordance with a preferred embodiment of the present invention, a limited bandwidth user first receives the first row of images and is immediately able to interact therewith. Over time, depending on the bandwidth available

Google Exhibit No. 1008
Page 192

to that user, successive data blocks are received, each cumulatively enhancing the quality. It is a particular feature of the invention that during receipt of successive blocks of data, the user is able to fully interact with the images.

Reference is now made to Fig. 5 showing a production tool 71 which accepts as an input a sequence of original digital frame units 72, integrated into a digital multimedia file 73. The production tool 71 includes an encoder unit 74 which operates by partitioning and compressing the digital multimedia file 73 into a scalable progressive database 75 comprised of data blocks 76. Successive blocks combine together to form higher bandwidth versions *a* to *n* of the media. Database 75 is stored on server 77. The production tool 71 enables the producer to control the bandwidth or quality granularity through control parameters 78. These parameters are used to calculate the data block sizes and compression settings within encoder unit 74.

Reference is now made to Fig. 6 showing the structure of data block 76 in scalable progressive database 75. Where a random access server is available, selective encoded frames 72 from block 76 are accessed at 79 on the server 77 database, based on interactive requests coming from the client. The encoded frames are transmitted from server 77 to a client computer 80 and integrated within client database 81, to mirror server database 75. It is appreciated that both sequential and random access servers may be advantageously employed in the present invention, although random access servers are preferred.

Reference is now made to Fig. 7 showing the decoder on a client computer 80. The client computer 80 receives from server 77 (Fig. 5) into a buffer 82 a series of data blocks 76 from scalable database 75. As the blocks are received, a client database, which mirrors the server database, is built up. A decoder unit 83 decompresses the blocks and an accumulator unit 84 integrates them to form a suitable low quality version 85 of the multimedia file 73 (Fig. 5), which is stored in a buffer 86, thus building up an interactive database.

The operations of the decoder 83 and accumulator 84 are governed by a CPU 87. They may operate in either order; i.e., the decoding may be carried out before the accumulation, or the accumulation may be carried out before the decoding. The multimedia file 73 is played on a player unit 88 in response to interactive user commands.

Google Exhibit No. 1008
Page 193

As the user interactively requests specific frames to be played, the buffer 86 supplies the highest quality version which it possesses. If the desired frame is not available, the buffer 86 sends back a request to buffer 82 to decode and accumulate that frame. If the frame is also not available in buffer 82, then that buffer 82 sends back a request to the server database 75 to transmit the frame. As playback continues and the bandwidth frees, additional data blocks 76 are received and integrated with the previously received blocks into higher quality versions 89 of the multimedia file.

The description of Figs. 1 - 7 has been directed towards the overall system and method provided by the present invention. The description which follows is directly principally to particular applications of the system and method described hereinabove.

Reference is now made to Fig. 8 showing a progressive database 75 for a digital video file which corresponds to digital multimedia file 73 (Fig. 5). The progressive database 75 comprises data blocks, where the first ones of data blocks 76 are used to create a preview 90 of the video, and the second ones of data blocks 76 are accumulated with the first blocks to create a full view 91 of the video. The views are stored in the interactive database buffer 86 and played in player unit 88, in response to interactive user commands.

Reference is now made to Fig. 9 showing a system for incorporating a progressive scalable database into a time-based video frame sequence with a macro and micro time scale, such as the one used in Apple's QUICKTIME® movies in accordance with a preferred embodiment of the present invention. Individual frames 72 are arranged according to a macro time scale, denoted by major axis markings 92 in Fig. 9. Each frame is displayed at the respective macro times indicated by markings 92.

Between successive frames 72, small fluctuations can be introduced. For example, the major time scale can be displaying a bird flying, and the minor time scale can be used to add fluttering to the bird's wings. The fluctuations typically involve only a small portion of the image area, and are displayed in rapid succession, according to minor axis markings 93 in Fig. 9. Such a time based sequence allows the decoder, which does the intensive processing to supply the frames 72, to run at a slow rate; e.g. 3 fps, whereas the viewer, doing the less intensive processing, can be playing at a fast rate; e.g. 30 fps. The fluctuations must be simple enough, though, that they can be rendered at the full 30

Google Exhibit No. 1008
Page 194

fps rate.

The present invention includes a novel use of such a two-scale time based system, to enable progressive streaming. In order to utilize the progressive database of the present invention in viewing such time-based video sequences, the minor time scale 93 is used to display progressive versions of a frame, rather than to display fluctuations, as originally conceived in the prior art. At each time corresponding to a minor axis marking, the player displays the latest version of the frame which is available. For example, in Fig. 9 the low quality version 94 of frame #4, corresponding to the first block 76, is displayed over a duration of three minor axis marks, by which time the second block 76 has been accumulated to form the medium quality version 95. This medium quality version is then displayed over a duration of two minor axis marks, by which time the third block 76 has been accumulated to form the high quality version 96 of the media. This high quality version is then displayed for a duration of three further minor axis marks, following which the frame advances to frame #5.

The cycle then repeats for progressive display of frame #5. If additional blocks arrive and are accumulated for frame #4, they are displayed when the video sequence is replayed. Thus it is seen that the progressive dimension of the database can be incorporated within the interactive dimension, through the use of a minor time scale, which is situated within the major time scale used for advancing the frames. The major time scale is the interactive axis, and the minor time scale becomes the progressive axis. The overall accomplishment is to enable viewing of the video before the full media stream has been downloaded. Initially, the low quality frames are displayed, and while additional blocks are downloaded and accumulated in background, the quality of the frames steadily improves.

In accordance with a preferred embodiment of the present invention, for the user to be able to view the video immediately, without waiting for the entire file to download, a production tool must store the encoded video in the order of successive blocks. Once the first block is downloaded, the video can already be viewed. The order of the encoded data items must therefore be:

Frame 1/Block 1, Frame 2/Block 1, ..., Frame *m* /Block 1,...,
Frame 1/Block *n*, Frame 2/Block *n*, ..., Frame *m* /Block *n*

Google Exhibit No. 1008
Page 195

Each row in this order comprises one entire block. In order to integrate this file format within the two-scale time-based system, each partial frame must be handled as if it were an entire frame. That is, the production tool must treat the movie as if there were a total of $m \cdot n$ distinct frames being encoded. The player in turn, however, must know that although it is receiving what appears to be $m \cdot n$ encoded frames, there are really only a total of m frames. It must decode and accumulate every successive sequence of m data items (i.e., each row of the above sequence) with the previous ones, to update the frames.

On the other hand, the natural order in which the production tool produces the encoded data items is:

Frame 1/Block 1, Frame 1/Block 2, ..., Frame 1/Block n, ...,

Frame m /Block 1, Frame m /Block 2, ..., Frame m /Block n

This is because each frame is sent only once to the codec for encoding, and returned as a series of encoded partial frames. Thus it is necessary to post-process the encoded data file, to rearrange the data items from this latter frame dominated order to the former block dominated order. This rearrangement process is referred to as "flattening" in the art, and is illustrated in Fig. 10.

As seen in Fig. 10, the frames 72 are supplied in sequence to the production tool, which produces a series of partial blocks 76 for each frame. The partial blocks are reordered, as indicated by the mapping in Fig. 10, into a single file stream 97. As indicated, the first partial blocks of each frame form the first m data units in the file 97, the second set of partial blocks of each frame form the next m data units, etc.

Reference is now made to Fig. 11 which shows a proxy system 98 for caching in a central hub 99 multimedia data which streams from servers 77 to clients 80. The servers 77 store their multimedia data in the progressive scalable representation described above in the server database 75, with each media encoded into data blocks 76 $B_1, B_2, \ldots, B_n$. Multiple servers 77 are connected through the central hub 99 to clients 80 via data communication channels. These channels are of various bandwidths. Whenever a client 80 issues a request for data from a specific server, a proxy computational unit 100 first determines, based on the server bandwidth, which data blocks are to be transmitted to the client. If those data blocks are not already cached in the hub, then the proxy 98

Google Exhibit No. 1008
Page 196

retrieves the required blocks 76 from the server along low bandwidth communication channels 101, at the appropriate bandwidth, and delivers them to the client 80 along high bandwidth communication channels 102. The blocks 76 are also cached in the hub 99.

There is provided an inventory flag database 103 on the hub which keeps a record of which data blocks are available. If some of the required data blocks 76 are already cached, then the proxy computational unit 100 computes which data blocks must be delivered from the server 77 in order to transmit to the client the highest quality version of the media possible, within the bandwidth constraint. The proxy 98 then retrieves the required blocks 76 from the server 77 along low bandwidth communication channels 101, stores them in its cache and delivers them to the client 80 along high bandwidth communication channels 102. A decoder unit 83 on the client computer decodes the data blocks received, and an accumulator unit 84 integrates them. The proxy 98 may also have its own decoder unit 83 and accumulator unit 84, which converts the data from its original compressed form on the server database 77 to an intermediate compressed form on the hub database 104, one which is faster to decompress than that of the client database. The proxy accumulation unit 84 may perform the necessary data block accumulation to store on the hub all possible versions of the multimedia in its intermediate compressed form, in which case the client accumulator 84 unit is unnecessary.

An update communication line 105 links servers 77 to the proxy 98, through which servers 77 can notify proxy 98 if any of the multimedia files have been updated. If proxy 98 receives such notification, then it clears its cache of any data blocks associated with those updated files, and resets its inventory flags 103, so that upon future client requests it will know that it has to retrieve the updated files from the server again.

There are several ways to generate the data blocks $B_1, B_2, \ldots, B_n$ for the progressive scalable database, so as to satisfy the user-selected constraints that

(*i*) for each $1 \leq k \leq n$, the first consecutive compressed data blocks $B_1, B_2, \ldots, B_k$ when integrated together produce a version of the media at a quality level commensurate with bandwidth f_k ;

(*ii*) the sizes of the compressed data blocks are such that the first consecutive compressed data blocks $B_1, B_2, \ldots, B_k$ when transmitted at bandwidth f_k suffice to enable

Google Exhibit No. 1008
Page 197

on-line playback of the media version.

Reference is now made to Fig. 12 which shows a system for generating a progressive scalable database from digital media data by cascading compressors in tandem. The user selects bit-rate and quality control parameters 106 for the encoding. The original multimedia data file 73 is input to a compressor 107 along with the user-selected control parameters 106, resulting in compressed data 108 adapted to a user-selected bandwidth. The compressed data 108 is transmitted to the scalable database 75 as the first data block 76. It is also transmitted to decompressor 109, which reconstructs the media as it would be generated on the client side.

The reconstructed data 110 is subtracted from the original data 73 to arrive at a residual 111. The residual 111 is fed back to the compressor 107 in a feedback loop 112, and compressed by compressor 107 with bit-rate control so that the compressed data 108 is adapted to the difference between the first and second user-selected bandwidths. The compressed data is transmitted to the progressive scalable database 75 as the second data block 76, and the loop continues repeatedly until the user-selected final quality is achieved.

It is not necessary for the compressor 107 to use the same compression method each time it operates. Rather, it can use a block identifier 113 as a parameter for switching between methods. For example, with a video encoder, the first block could be encoded using a low quality version of H.263 and successive blocks could be encoded using spatial vector quantization and temporal wavelets.

One possible approach to compression is the use of fractal technology, such as that described and claimed in applicant's U.S. Patent No. 5,497,435, the disclosure of which is hereby incorporated by reference.

Another approach to generating a scalable database is through the use of progressive JPEG. The progressive JPEG standard allows the encoder to segment the compression into spectral selection and successive approximation scans. In spectral selection the DCT coefficients are grouped into spectral bands, and in successive approximation the bits used to represent them are divided into lower and higher precision information. Progressive JPEG is described by Pennebaker, W. B. and Mitchell, J. L. in *JPEG: Still Image Data Compression*, Van-Nostrand Reinhold, New York, 1993, the

Google Exhibit No. 1008
Page 198

disclosure of which is hereby incorporated by reference.

Reference is now made to Fig. 13 which shows a decoder for the scalable database 75 on the client side, corresponding to the encoder from Fig. 12. Blocks 76 are successively transmitted from the server to the client and are decompressed in decoder 83. The decoded data is integrated in an accumulator 84, which converts it into a form compatible with the player 88, and stores it in the interactive buffer 86. While the player is showing the media, additional blocks 76 are received in the background, decoded and accumulated, so that the media quality is upgraded when it is replayed. The player 88 requests frames from buffer 86, and if they are not available then buffer 86 requests them from database 75. A block identifier 113 provides an input to the decoder as each block is decoded, so that the decoder can apply the appropriate decompression method for that data block, corresponding to the compression method which was performed in compressor 107.

Reference is now made to Fig. 14 which shows a progressive database for a large still image. The first encoded frame unit of a data block consists of the full image sub-sampled 4:1 in each dimension. The second set of frames consist of four encoded frame units corresponding to each of the quadrants of the full image, sub-sampled at 2:1. Although this second set of frames appears to be four times the size of the first frame, it can be stored using only three times as much data since the decoder can also accumulate data from the previously displayed frame. The third set of frames includes sixteen encoded frame units, each corresponding to a quadrant of a quadrant of the full image, but at the original scale. Again, this third set of frames stores three times as much data as the second set. These three sets of frames comprise the entire encoded data block, and their sum total is, of course, the same amount of data as the full image at the original scale. (The effect of the compression is being ignored here.) The frames are all arranged sequentially in the encoded data block 76. The mapping from multi-resolution image tiles to sequential frames 72 is shown in Fig. 14.

When the frame #1 is downloaded, decoded and displayed, the viewer sees a low resolution version of the full image. In the example shown in Fig. 14, the viewer clicks on a hot spot in the northwest quadrant and link 114 then transfers over to the northwest frame in the second set of frames, which corresponds to frame #2 in Fig. 14. Displaying

Google Exhibit No. 1008
Page 199

this new frame gives the appearance of having zoomed in on the quadrant. The viewer next clicks on a hot spot in the southeast quadrant of the frame being displayed from the second set, and link 114 then transfers over to the indicated frame in the third set of frames, which corresponds to frame #11 in the Fig. 14, giving the effect of yet another zoom in.

While the viewer is looking at the first frame, the data from the second set of frames is being downloaded in the background. Once received, this data is decoded and stored in the client interactive database. When the hot spot in the northwest quadrant is clicked, the client CPU looks for the desired frame #2 in its interactive database. If the frame is already present, it is delivered to the viewer for immediate display; otherwise the interactive buffer sends the request back to the progressive database. The progressive database can access and send the specific encoded frame required from the second set of frames, without sending all of the frames. Finally, as successive blocks are integrated into the interactive database, the quality of the individual frames is enhanced, and the zoomed in tile looks progressively better and better, having cumulatively enhanced quality.

It is evident to those skilled in the art that the above discussion applies to animations and panoramas of large images which contain areas of interest such as sprites. The frames comprising the interactivity dimension of the database would correspond to multi-resolution tiles from each of the individual images in the animation or panorama.

Fig. 15 illustrates a system for transmitting VRML images over a server/client network. A VRML database 121 is stored on a server. A client 80 interactively controls the VRML viewing parameters 122 through use of a mouse and keyboard 23. The viewing parameters 122 are used in conjunction with the VRML database 121 to render a raster bitmap image 123 of the VRML object on the server computer. If the bitmap image corresponding to the viewing parameters was already rendered previously, then the frame data for the bitmap in the client database 42 is used for display by the player 88. The raster bitmap is encoded into partial frames by encoder 74, and the partial frames are inserted into a server two-dimensional database 41. The server database is continually streamed to the client, building up the client database 42. The client database 42 is used to provide the frames for display.

Google Exhibit No. 1008
Page 200

Fig. 16 depicts a flowchart for the VRML application shown in Fig. 15. At step 131, the user interacts with the mouse and keyboard, and the client computer updates the viewing parameters. At step 132 the client computer checks whether or not those viewing parameters have already been processed. Although the viewing parameters can vary continuously, a preferred embodiment of the present invention discretizes them to a finite number of settings; for example, 10° resolution for angles. This makes it likely that the user will navigate back to the same settings used earlier.

If the viewing parameters are new, then they are sent to the server computer, which renders the VRML database into a raster bitmap corresponding to the specific viewing parameters selected, at step 133. At step 134 the bitmap is encoded into partial frames, and incorporated into the server database. Step 135 is continually operative to transmit additional encoded data from the server to the client. As data is received on the client, the client database is built up at step 136. At step 137 the client database generates the latest version of the bitmap on demand, and displays it. This step is also carried out whenever step 132 results in confirmation that the viewing parameters have already been processed.

It is evident to those skilled in the art that the above discussion applies to CAD/CAM models for three-dimensional objects, as well as VRML models. Specifically, Fig. 17 illustrates a typical two-dimensional array of CAD/CAM images, indexed vertically according to progressive coordinate and horizontally according to interactive coordinate. It is appreciated that the images in a given horizontal row may be viewed interactively. Each successive horizontal row of images is built up over time at a rate determined by bandwidth availability and has increased quality inasmuch as it is based on an increasing number of data blocks.

It will be appreciated by persons skilled in the art that the present invention is not limited by what has been particularly shown and described hereinabove but extends also to embodiments which would naturally occur to persons reading the above description and to combinations and subcombinations of embodiments described hereinabove.

Google Exhibit No. 1008
Page 201

CLAIMS

1. A system for transmitting digital data representing an original over plural transmission links, at least some of which have limited bandwidth, comprising:

a digital data source storing digital data representing the original;

a digital data receiver receiving said digital data representing the original via one of said plural transmission links having limited bandwidth; and

a digital data transmitter operative to transmit said digital data representing the original to said receiver over a transmission link having a limited bandwidth in plural blocks which are sequentially transmitted at a rate determined by said limited bandwidth, each block being an incomplete collection of data which includes parts of multiple frames, each frame being viewable in a selectable order by said receiver even when less than all of said plural blocks have been received, receipt of subsequent blocks by the receiver being used to cumulatively improve the quality of the digital data viewed by the receiver.

2. A system according to claim 1 and wherein a first one of said plural blocks contains digital data which represents a first approximation to said original.

3. A system according to claim 2 and wherein additional ones of said plural blocks, when combined with said first one of said plural blocks provide additionally accurate approximations to said original.

4. A system according to claim 1 and wherein each of said multiple frames comprises a portion of data which can be independently and interactively manipulated.

5. A system according to claim 1 and also comprising a block generator operative to receive digital data representing said original and to provide said plural blocks.

Google Exhibit No. 1008
Page 202

6. A system according to claim 5 and wherein said block generator is operative to provide plural blocks which are distinguished from each other by their respective frequency bands.

7. A system according to claim 5 and wherein said block generator comprises a fractal compression engine.

8. A system according to claim 7 and wherein said block generator is operative to decompose said digital data representing the original in relatively high frequency and relatively low frequency digital data portions, and wherein a first plurality of blocks containing said relatively low frequency portion is transmitted by said data transmitter prior to transmission of a second plurality of blocks containing said relatively high frequency portion.

9. A system according to claim 5 and wherein said block generator is operative to provide plural blocks by sampling said received digital data.

10. A system according to claim 9 and wherein the sampling rate of a plurality of blocks is equal to the sum of the sampling rates of individual ones of said plurality of blocks.

11. A system according to claim 6 and wherein said block generator comprises a wavelet encoder.

12. A system according to claim 5 and wherein said block generator comprises a quantizer which produces blocks each of which contain quantized data of a different order, such that accumulation of multiple blocks provides combined data of greater precision than that contained in any single block.

13. A block generator comprising:

a producer interface; and

Google Exhibit No. 1008
Page 203

a digital data compressor, operative in response to producer control parameters received via said producer interface for receiving digital data representing the original and providing plural blocks, each block being an incomplete collection of data which includes parts of multiple frames.

14. A block generator according to claim 13 and being operative to provide plural blocks which are distinguished from each other by their respective frequency bands.

15. A block generator according to claim 13 and comprising a fractal compression engine.

16. A block generator according to claim 15 and being operative to decompose said digital data representing the original in relatively high frequency and relatively low frequency digital data portions, and wherein a first plurality of blocks containing said relatively low frequency portion is transmitted by said data transmitter prior to transmission of a second plurality of blocks containing said relatively high frequency portion.

17. A block generator according to claim 13 and being operative to provide plural blocks by sampling said received digital data.

18. A block generator according to claim 17 and wherein the sampling rate of a plurality of blocks is equal to the sum of the sampling rates of individual ones of said plurality of blocks.

19. A block generator according to claim 14 and comprising a wavelet encoder.

20. A block generator according to claim 13 and comprising a quantizer which produces blocks each of which contain quantized data of a different order, such that accumulation of multiple blocks provides combined data of greater precision than that contained in any single block.

Google Exhibit No. 1008
Page 204

21. A block generator according to claim 15 and comprising a quantizer which produces blocks each of which contain quantized data of a different order, such that accumulation of multiple blocks provides combined data of greater precision than that contained in any single block.

22. A block generator according to claim 17 and comprising a quantizer which produces blocks each of which contain quantized data of a different order, such that accumulation of multiple blocks provides combined data of greater precision than that contained in any single block.

23. A block generator according to claim 20 and comprising a quantizer which produces blocks each of which contain quantized data of a different order, such that accumulation of multiple blocks provides combined data of greater precision than that contained in any single block.

24. A digital data receiver comprising:

a data receipt interface receiving digital data representing an original in a plurality of sequential blocks, each block being an incomplete collection of data which includes parts of multiple frames;

a block accumulator for combining plural blocks as they are received for viewing by the recipient; and

a viewer including a recipient interface which permits each frame to be viewed in an order selected by the recipient, even when less than all of said plural blocks have been received, combining of plural blocks by said block accumulator being used to improve the quality of the digital data viewed by said recipient.

25. A digital data receiver according to claim 24 and wherein said block accumulator is operative to combine plural blocks which are distinguished from each other by their respective frequency bands.

Google Exhibit No. 1008
Page 205

26. A digital data receiver according to claim 24 and comprising a fractal decompression engine.

27. A digital data receiver according to claim 26 and wherein said data receipt interface is operative to initially receive a first plurality of blocks containing relatively low frequency data and thereafter receive a second plurality of blocks containing relatively high frequency data and wherein said block accumulator is operative to reconstitute said digital data representing the original from said blocks representing relatively high frequency and relatively low frequency data.

28. A digital data receiver according to claim 24 and wherein said block accumulator is operative to combine plural blocks having different sampling.

29. A digital data receiver according to claim 28 and wherein the sampling rate of a combined plurality of blocks is equal to the sum of the sampling rates of individual ones of said plurality of blocks.

30. A digital data receiver according to claim 25 and comprising a wavelet decoder.

31. A digital data receiver according to claim 24 and wherein said block accumulator comprises a dequantizer which combines blocks each of which contain quantized data of a different order, such that accumulation of multiple blocks provides combined data of greater precision than that contained in any single block.

32. A digital data receiver according to claim 26 and wherein said block accumulator comprises a dequantizer which combines blocks each of which contain quantized data of a different order, such that accumulation of multiple blocks provides combined data of greater precision than that contained in any single block.

33. A digital data receiver according to claim 30 and wherein said block accumulator comprises a dequantizer which combines blocks each of which contain quantized data of

Google Exhibit No. 1008
Page 206

a different order, such that accumulation of multiple blocks provides combined data of greater precision than that contained in any single block.

34. A digital data transmitter actuator comprising:

an organizer operative, when actuated, to access digital data representing the original which is organized in plural blocks for subsequent transmission, each block being an incomplete collection of data which includes parts of multiple frames, each frame being viewable in a selectable order by said a receiver even when less than all of said plural blocks have been received; and

a receiver instruction interface responsive to interactive inputs from a receiver for actuating said organizer to select a given block and at least one given partial frame within said given block for transmission.

35. A method for transmitting digital data representing an original over plural transmission links at least some of which have limited bandwidth comprising:

storing digital data representing the original;

receiving at a receiver said digital data representing the original via one of said plural transmission links having limited bandwidth; and

transmitting said digital data representing the original to said receiver over a transmission link having a limited bandwidth in plural blocks which are sequentially transmitted at a rate determined by said limited bandwidth, each block being an incomplete collection of data which includes parts of multiple frames, each frame being viewable in a selectable order by said receiver even when less than all of said plural blocks have been received, receipt of subsequent blocks by the receiver being used to cumulatively improve the quality of the digital data viewed by the receiver.

36. A method according to claim 35 and wherein a first one of said plural blocks contains digital data which represents a first approximation to said original.

Google Exhibit No. 1008
Page 207

37. A method according to claim 36 and wherein additional ones of said plural blocks, when combined with said first one of said plural blocks provide additionally accurate approximations to said original.

38. A method according to claim 35 and wherein each of said multiple frames comprises a portion of data which can be independently and interactively manipulated.

39. A method according to claim 35 and also comprising the step of block generation operative to receive digital data representing said original and to provide said plural blocks.

40. A method according to claim 39 and wherein said block generation is operative to provide plural blocks which are distinguished from each other by their respective frequency bands.

41. A method according to claim 39 and wherein said block generation employs fractal compression.

42. A method according to claim 41 and wherein said block generation includes decomposing said digital data representing the original in relatively high frequency and relatively low frequency digital data portions, and wherein a first plurality of blocks containing said relatively low frequency portion is transmitted prior to transmission of a second plurality of blocks containing said relatively high frequency portion.

43. A method according to claim 39 and wherein said block generation provides plural blocks by sampling said received digital data.

44. A method according to claim 43 and wherein the sampling rate of a plurality of blocks is equal to the sum of the sampling rates of individual ones of said plurality of blocks.

Google Exhibit No. 1008
Page 208

45. A method according to claim 40 and wherein said block generation comprises wavelet encoding.

46. A method according to claim 39 and wherein said block generation comprises quantizing into blocks each of which contain quantized data of a different order, such that accumulation of multiple blocks provides combined data of greater precision than that contained in any single block.

47. A method for digital data transmission comprising:

organizing digital data representing the original into plural blocks for subsequent transmission, each block being an incomplete collection of data which includes parts of multiple frames, each frame being viewable in a selectable order by said a receiver even when less than all of said plural blocks have been received;

responsive to interactive inputs from a receiver for actuating said organizer, selecting a given block and at least one given partial frame within said given block for transmission; and

transmitting the selected given block and at least one given partial frame to a user.

48. A method for providing on-line virtual reality movies, comprising:

inputting a cyclic movie sequence into an encoder; determining the number of portions that each frame of said movie is divided into, and forming partial frames;

specifying hot-spots and independent objects for interaction within a partial frame;

transmitting the partial frames part by part to a user's asynchronous database, and

displaying said frames on a user's interface.

49. The method as claimed in claim 48, wherein said partial frames are generated by forming a partial resolution frame and subtracting same from the original frame from which it was formed.

50. The method as claimed in claim 48, wherein said partial frames are generated by repeating the process several times.

Google Exhibit No. 1008
Page 209

51. The method as claimed in claim 48, wherein said partial frames are compression encoded prior to being transmitted and compression decoded prior to being displayed on the user's interface.

52. A system for providing on-line virtual reality movies, comprising:

a production workstation for producing VR movies;

an encoder for preparing the VR movie for transmission; and

a server including a repository for the VR movie and a transceiver for transmitting the movie part by part to a user, upon request.

53. The system as claimed in claim 52, wherein said encoder comprises a partial frame generator and a controller.

54. The system as claimed in claim 52, wherein the encoder further comprises a compressor for compression encoding image parts produced in said workstation.

55. The system as claimed in claim 52, further comprising

a user's transceiver for receiving VR movie parts transmitted by said server;

an asynchronous database for supplying the user with images upon request, independent of transmission, and a decoder for integrating partial frames and facilitating a progressively updated display of images; and

a user's workstation for interaction with the VR movie.

56. The system as claimed in claim 55, wherein the decoder further comprises a decompressor and a partial frame integrator.

57. A method for providing on-line virtual reality movies, according to claim 48, substantially as hereinbefore described and with reference to the accompanying drawings.

Google Exhibit No. 1008
Page 210

58. A system for providing on-line virtual reality movies, according to claim 52, substantially as hereinbefore described and with reference to the accompanying drawings.

59. A method for encoding original digital video data to be stored on a server computer for on-line delivery to client computers, comprising the steps of:

encoding the digital video into a database comprising a series of encoded data blocks, each block comprising a sequence of encoded frames, with the property that successive blocks when decoded and integrated together provide successively higher bandwidth versions of the video for on-line playback;

storing the database on a server computer;

processing a request by a client computer for on-line delivery of the video in order to determine which data blocks to transmit, so as to accommodate the client bandwidth;

transmitting the necessary data blocks to the client;

decoding the data blocks on the client computer;

integrating the data blocks together on the client computer to reconstruct an appropriate version of the original digital video; and

playing the reconstructed video on the client computer.

60. A method according to Claim 59 wherein said step of encoding includes a bit-rate control device enabling the producer to pre-select the sequence of bandwidths or quality levels for the database.

61. A method according to Claim 59 wherein said step of encoding is performed in such a way that the first blocks of the database correspond to previews of the video.

62. A method according to Claim 59 wherein said steps of transmitting, decoding, integrating and playing are repeated in succession a number of times in order to transmit additional data blocks to the client, thereby upgrading the quality of the video while it is replayed.

Google Exhibit No. 1008
Page 211

63. A method for encoding original digital audio data to be stored on a server computer for on-line delivery to client computers, comprising the steps of:

encoding the digital audio into a database comprising a series of encoded data blocks, each block comprising a sequence of encoded frames, with the property that successive blocks when decoded and integrated together provide successively higher bandwidth versions of the audio for on-line playback;

storing the database on a server computer;

processing a request by a client computer for on-line delivery of the audio in order to determine which data blocks to transmit, so as to accommodate the client bandwidth;

transmitting the necessary data blocks to the client;

decoding the data blocks on the client computer;

integrating the data blocks together on the client computer to reconstruct an appropriate version of the original digital audio; and

playing the reconstructed audio on the client computer.

64. A method according to Claim 63 wherein said step of encoding includes a bit-rate control device enabling the producer to pre-select the sequence of bandwidths or quality levels for the database.

65. A method according to Claim 63 wherein said steps of transmitting, decoding, integrating and playing are repeated in succession a number of times in order to transmit additional data blocks to the client, thereby upgrading the quality of the audio while it is replayed.

66. A method for encoding original digital object movie data to be stored on a server computer for on-line delivery to client computers, comprising the steps of:

encoding the digital object movie into a database comprising a series of encoded data blocks, each block comprising a sequence of encoded frames, with the property that successive blocks when decoded and integrated together provide successively higher bandwidth versions of the object movie for on-line playback;

storing the database on a server computer;

Google Exhibit No. 1008
Page 212

processing a request by a client computer for on-line delivery of the object movie in order to determine which data blocks to transmit, so as to accommodate the client bandwidth;

transmitting the necessary data blocks to the client;

decoding the data blocks on the client computer;

integrating the data blocks together on the client computer to reconstruct an appropriate version of the original digital object movie; and

playing the reconstructed object movie on the client computer.

67. A method according to Claim 66 wherein said step of encoding includes a bit-rate control device enabling the producer to pre-select the sequence of bandwidths or quality levels for the database.

68. A method according to Claim 66 wherein said steps of transmitting, decoding, integrating and playing are repeated in succession a number of times in order to transmit additional data blocks to the client, thereby upgrading the quality of the object movie while it is replayed.

69. A method for encoding an original digital panorama to be stored on a server computer for on-line delivery to client computers, comprising the steps of:

encoding the digital panorama into a database comprising a series of encoded data blocks, each block comprising a sequence of encoded frames, with the property that successive blocks when decoded and integrated together provide successively higher bandwidth versions of the panorama for on-line playback;

storing the database on a server computer;

processing a request by a client computer for on-line delivery of the panorama in order to determine which data blocks to transmit, so as to accommodate the client bandwidth;

transmitting the necessary data blocks to the client;

decoding the data blocks on the client computer;

Google Exhibit No. 1008
Page 213

integrating the data blocks together on the client computer to reconstruct an appropriate version of the original digital panorama; and

playing the reconstructed panorama on the client computer.

70. A method according to Claim 69 wherein said step of encoding includes a bit-rate control device enabling the producer to pre-select the sequence of bandwidths or quality levels for the database.

71. A method according to Claim 69 wherein said steps of transmitting, decoding, integrating and playing are repeated in succession a number of times in order to transmit additional data blocks to the client, thereby upgrading the quality of the panorama while it is replayed.

72. A method for encoding original digital large still image data to be stored on a server computer for on-line delivery to client computers, comprising the steps of:

encoding the large digital image into a database comprising a series of encoded data blocks, each block comprising a sequence of encoded multi-resolution tiles of the image, with the property that successive blocks when decoded and integrated together provide successively higher quality versions of the tiles for display;

storing the database on a server computer;

processing a request by a client computer for on-line delivery of the image in order to determine which data blocks to transmit;

transmitting the necessary data blocks to the client;

decoding the data blocks on the client computer;

integrating the data blocks together on the client computer to reconstruct an appropriate version of the original multi-resolution image tiles; and

interactively displaying the reconstructed tiles on the client computer.

73. A method according to Claim 72 wherein said step of encoding includes a compression control device enabling the producer to pre-select the sequence of quality levels for the database.

Google Exhibit No. 1008
Page 214

74. A method according to Claim 73 wherein said step of encoding operates on a plurality of images forming an animation, and each encoded data block is comprised of multi-resolution tiles from the plurality of images.

75. A video processing system operative on digital video data for encoding the digital video, storing it on a server computer and delivering it to client computers on-line upon request comprising:

an encoder for compressing the digital video into a database comprising a series of encoded data blocks, each block comprising a sequence of encoded frames, with the property that successive blocks when decoded and integrated together provide successively higher bandwidth versions of the video for on-line playback;

a storage device for archiving the database on a server computer;

a processing unit for accepting a request by a client computer for on-line delivery of the video and determining which data blocks to transmit, so as to accommodate the client bandwidth;

a transmitter for delivering the necessary data blocks to the client;

a decoder for decompressing the data blocks back into video data on the client computer;

an accumulator for integrating the data blocks together on the client computer to reconstruct an appropriate version of the original digital video; and

a player on the client computer for playing the reconstructed digital video.

76. The system as claimed in Claim 75 wherein said encoder includes a bit-rate controller enabling the user to pre-select the sequence of bandwidths or quality levels for the database.

77. The system as claimed in Claim 75 wherein said encoder compresses the digital video in such a way that the first blocks of the database correspond to previews of the video.

Google Exhibit No. 1008
Page 215

78. The system as claimed in Claim 75 wherein the transmitter, decoder, accumulator and player repeatedly operate in succession a number of times in order to transmit additional data blocks to the client, thereby upgrading the quality of the video while it is being replayed.

79. An audio processing system operative on digital audio data for encoding the digital audio, storing it on a server computer and delivering it to client computers on-line upon request comprising:

an encoder for compressing the digital audio into a database comprising a series of encoded data blocks, each block comprising a sequence of encoded frames, with the property that successive blocks when decoded and integrated together provide successively higher bandwidth versions of the audio for on-line playback;

a storage device for archiving the database on a server computer;

a processing unit for accepting a request by a client computer for on-line delivery of the audio and determining which data blocks to transmit, so as to accommodate the client bandwidth;

a transmitter for delivering the necessary data blocks to the client;

a decoder for decompressing the data blocks back into audio data on the client computer;

an accumulator for integrating the data blocks together on the client computer to reconstruct an appropriate version of the original digital audio; and

a player on the client computer for playing the reconstructed digital audio.

80. The system as claimed in Claim 79 wherein said encoder includes a bit-rate controller enabling the user to pre-select the sequence of bandwidths or quality levels for the database.

81. The system as claimed in Claim 79 wherein the transmitter, decoder, accumulator and player repeatedly operate in succession a number of times in order to transmit additional data blocks to the client, thereby upgrading the quality of the audio while it is being replayed.

Google Exhibit No. 1008
Page 216

82. An object movie processing system operative on digital object movie data for encoding the digital object movie, storing it on a server computer and delivering it to client computers on-line upon request comprising:

an encoder for compressing the digital object movie into a database comprising a series of encoded data blocks, each block comprising a sequence of encoded frames, with the property that successive blocks when decoded and integrated together provide successively higher bandwidth versions of the object movie for on-line playback;

a storage device for archiving the database on a server computer;

a processing unit for accepting a request by a client computer for on-line delivery of the object movie and determining which data blocks to transmit, so as to accommodate the client bandwidth;

a transmitter for delivering the necessary data blocks to the client;

a decoder for decompressing the data blocks back into object movie data on the client computer;

an accumulator for integrating the data blocks together on the client computer to reconstruct an appropriate version of the original digital object movie; and

a player on the client computer for playing the reconstructed digital object movie.

83. The system as claimed in Claim 82 wherein said encoder includes a bit-rate controller enabling the user to pre-select the sequence of bandwidths or quality levels for the database.

84. The system as claimed in Claim 83 wherein the transmitter, decoder, accumulator and player repeatedly operate in succession a number of times in order to transmit additional data blocks to the client, thereby upgrading the quality of the object movie while it is being replayed.

85. A panorama processing system operative on digital panorama data for encoding the digital panorama, storing it on a server computer and delivering it to client computers on-line upon request comprising:

Google Exhibit No. 1008
Page 217

an encoder for compressing the digital panorama into a database comprising a series of encoded data blocks, each block comprising a sequence of encoded frames, with the property that successive blocks when decoded and integrated together provide successively higher bandwidth versions of the panorama for on-line playback;

a storage device for archiving the database on a server computer;

a processing unit for accepting a request by a client computer for on-line delivery of the panorama and determining which data blocks to transmit, so as to accommodate the client bandwidth;

a transmitter for delivering the necessary data blocks to the client;

a decoder for decompressing the data blocks back into panorama data on the client computer;

an accumulator for integrating the data blocks together on the client computer to reconstruct an appropriate version of the original digital panorama; and

a player on the client computer for playing the reconstructed digital panorama.

86. The system as claimed in Claim 85 wherein said encoder includes a bit-rate controller enabling the user to pre-select the sequence of bandwidths or quality levels for the database.

87. The system as claimed in Claim 85 wherein the transmitter, decoder, accumulator and player repeatedly operate in succession a number of times in order to transmit additional data blocks to the client, thereby upgrading the quality of the panorama while it is being replayed.

88. An image processing system operative on large digital image data for encoding the digital image, storing it on a server computer and delivering it to client computers on-line upon request comprising:

an encoder for compressing the large digital image into a database comprising a series of encoded data blocks, each block comprising a sequence of encoded multi-resolution tiles of the image, with the property that successive blocks when decoded and integrated together provide successively higher quality versions of the image tiles;

Google Exhibit No. 1008
Page 218

a storage device for archiving the database on a server computer;

a processing unit for accepting a request by a client computer for on-line delivery of the image and determining which data blocks to transmit;

a transmitter for delivering the necessary data blocks to the client;

a decoder for decompressing the data blocks back into image tile data on the client computer;

an accumulator for integrating the data blocks together on the client computer to reconstruct an appropriate version of the original multi-resolution image tiles; and

an interactive viewer on the client computer for displaying the reconstructed image tiles.

89. The system as claimed in Claim 88 wherein said encoder includes a compression controller enabling the user to pre-select the sequence of quality levels for the database.

90. The system as claimed in Claim 88 wherein said encoder operates on a plurality of images forming an animation, and each encoded data block is comprised of multi-resolution tiles from the plurality of images.

91. A method for caching of data which gets transmitted from servers to clients on a central hub within a network, comprising the steps of:

encoding digital multimedia data into databases comprising a series of encoded data blocks, each block comprising a sequence of encoded frames, with the property that successive blocks when decoded and integrated together provide successively higher bandwidth versions of the media for on-line playback;

storing the databases on a multitude of server computers;

managing within the hub requests by client computers for on-line delivery of media stored on server computers in order to determine which data blocks to transmit, so as to accommodate the client bandwidth;

transmitting the necessary data blocks from the server and from the hub to the client;

Google Exhibit No. 1008
Page 219

storing the data blocks delivered by the server in the cache residing in the central hub;

processing within the hub the data blocks it receives;

decoding the data blocks on the client computer;

integrating the data blocks together on the client computer to reconstruct an appropriate version of the original digital media; and

playing the reconstructed media on the client computer.

92. A method according to Claim 91 wherein said step of managing is performed by:

setting inventory flags to indicate which data blocks are currently stored in the hub.

93. A method according to Claim 92 wherein said step of managing further includes the steps of:

communicating with the servers to monitor which media data is outdated;

removing from cache the blocks corresponding to the media data which is outdated; and

resetting the inventory flags to indicate that the above blocks are no longer stored in the cache.

94. A method according to Claim 91 wherein said step of processing comprises the steps of:

decoding the data blocks received;

integrating the data blocks together to reconstruct appropriate versions of the original digital media; and

encoding the reconstructed media versions into an intermediate database for future transmission to the clients.

95. A proxy system operative on a server/client network for caching of data which gets transmitted from servers to clients on a central hub, comprising:

an encoder for compressing digital multimedia data into databases comprising a series of encoded data blocks, each block comprising a sequence of encoded frames, with

Google Exhibit No. 1008
Page 220

the property that successive blocks when decoded and integrated together provide successively higher bandwidth versions of the media for on-line playback;

server communication lines from the servers to the hub for sending data blocks;

client communication lines from the hub to the clients for sending digital data;

storage devices for archiving the databases on a multitude of server computers;

a management unit within the hub to process requests by client computers for on-line delivery of media stored on server computers in order to determine which data blocks to transmit, so as to accommodate the client bandwidth;

a transmitter for delivering the necessary data blocks on the server communication lines from the server to the hub, and on the client communication lines from the hub to the client;

a storage device for saving the data blocks delivered by the server communication lines in the cache residing in the central hub;

a processing unit within the hub for processing the data blocks which the hub receives;

a decoder for decompressing the data blocks on the client computer;

an accumulator for integrating the data blocks together on the client computer to reconstruct an appropriate version of the original digital media; and

a player for playing the reconstructed media on the client computer.

96. The system as claimed in Claim 95 wherein said management unit operates by setting inventory flags to indicate which data blocks are currently stored in the hub.

97. The system as claimed in Claim 96 wherein said management unit operates by monitoring from the servers which media data is outdated, removing from cache the blocks corresponding to the media data which is outdated, and resetting the inventory flags to indicate that the above blocks are no longer stored in the cache.

98. The system as claimed in Claim 95 wherein said processing unit comprising:

a decoder for decompressing the data blocks received;

Google Exhibit No. 1008
Page 221

an accumulator for integrating the data blocks together to reconstruct appropriate versions of the original digital media; and

an encoder for compressing the reconstructed media versions into an intermediate database for future transmission to the clients.

99. A multi-casting unit (MCU) system operative on a broadcasting network for caching of data which gets transmitted from stations to viewers, comprising:

an encoder for compressing digital multimedia data into databases comprising a series of encoded data blocks, each block comprising a sequence of encoded frames, with the property that successive blocks when decoded and integrated together provide successively higher bandwidth versions of the media for on-line playback;

station communication lines from the stations to the MCU for sending data blocks;

viewer communication lines from the MCU to the viewers for sending data;

viewer receiver units for receiving the data sent by the MCU;

storage devices for archiving the databases on a multitude of station computers;

a management unit within the MCU to process requests by viewers for on-line delivery of media stored on station computers in order to determine which data blocks to transmit, so as to accommodate the viewer bandwidth;

a transmitter for delivering the necessary data blocks on the station communication lines from the station to the MCU, and on the viewer communication lines from the MCU to the viewer receiver units;

a storage device for saving the data blocks delivered by the station communication lines in the cache residing in the MCU;

a processing unit within the MCU for processing the data blocks which the MCU receives;

a decoder for decompressing the data blocks on the viewer receiver;

an accumulator for integrating the data blocks together on the viewer receiver unit to reconstruct an appropriate version of the original digital media; and

a player for playing the reconstructed media from the viewer receiver unit.

Google Exhibit No. 1008
Page 222

100. The system as claimed in Claim 99 wherein said management unit operates by setting inventory flags to indicate which data blocks are currently stored in the MCU.

101. The system as claimed in Claim 100 wherein said management unit operates by monitoring from the stations which media data is outdated, removing from cache the blocks corresponding to the media data which is outdated, and resetting the inventory flags to indicate that the above blocks are no longer stored in the cache.

102. The system as claimed in Claim 99 wherein said processing unit within the MCU comprising:

a decoder for decompressing the data blocks received;

an accumulator for integrating the data blocks together to reconstruct appropriate versions of the original digital media; and

an encoder for compressing the reconstructed media versions into an intermediate database for future transmission to the viewers.

103. A method for streaming multimedia data over a network, comprising the steps of:

encoding the media into a progressive database indexed according to frame and progressive block numbers;

serializing the encoded database;

storing the serialized database on a server;

streaming the serialized database to a client upon request;

creating a mirror copy of the encoded database on the client computer from the data which streams in; and

decoding the encoded database on the client computer into a sequence of frames for real time display.

104. A multimedia network streaming system, comprising:

an encoder for compressing the media into a progressive database indexed according to frame and progressive block numbers;

a sequencer for serializing the encoded database;

Google Exhibit No. 1008
Page 223

a storage device for archiving the serialized database on a server;

a transmitter for streaming the serialized database to a client upon request;

a processor for creating a mirror copy of the encoded database on the client computer from the data which streams in; and

a decoder for decompressing the encoded database on the client computer into a sequence of frames for real time display.

105. A system for transmitting model based data representations of three dimensional images over plural transmission links having limited bandwidth, said system comprising:

a digital data source storing model based data representations of three dimensional images;

an image processor for rendering views of said model based data representations into raster bitmap format;

a digital data receiver receiving said digital data in said raster bitmap format over a one of said plural transmission links having limited bandwidth; and

a digital data transmitter operative to transmit said digital data in said raster bitmap format to said receiver over a transmission link having a limited bandwidth in plural blocks which are sequentially transmitted at a rate determined by said limited bandwidth, each block being an incomplete collection of data which includes parts of multiple frames, each frame being viewable in a selectable order by said receiver even when less than all of said plural blocks have been received, receipt of subsequent blocks by the receiver being used to cumulatively improve the quality of the digital data viewed by the receiver.

106. A system according to claim 105 and wherein said model based data representations comprise VRML representations.

107. A system according to claim 105 and wherein said model based data representations comprise CAD-CAM representations.

Google Exhibit No. 1008
Page 224

108. A system according to claim 105 and wherein said image processor is operative to render only views which are selected by a user.

109. A method for transmitting model based data representations of three dimensional images over plural transmission links having limited bandwidth, said system comprising:

storing model based data representations of three dimensional images;

rendering views of said model based data representations into raster bitmap format;

receiving said digital data in said raster bitmap format over a one of said plural transmission links having limited bandwidth; and

transmit said digital data in said raster bitmap format to said receiver over a transmission link having a limited bandwidth in plural blocks which are sequentially transmitted at a rate determined by said limited bandwidth, each block being an incomplete collection of data which includes parts of multiple frames, each frame being viewable in a selectable order by said receiver even when less than all of said plural blocks have been received, receipt of subsequent blocks by the receiver being used to cumulatively improve the quality of the digital data viewed by the receiver.

110. A method according to claim 109 and wherein said model based data representations comprise VRML representations.

111. A method according to claim 109 and wherein said model based data representations comprise CAD-CAM representations.

112. A method according to claim 109 and wherein said image processor is operative to render only views which are selected by a user.

Google Exhibit No. 1008
Page 225

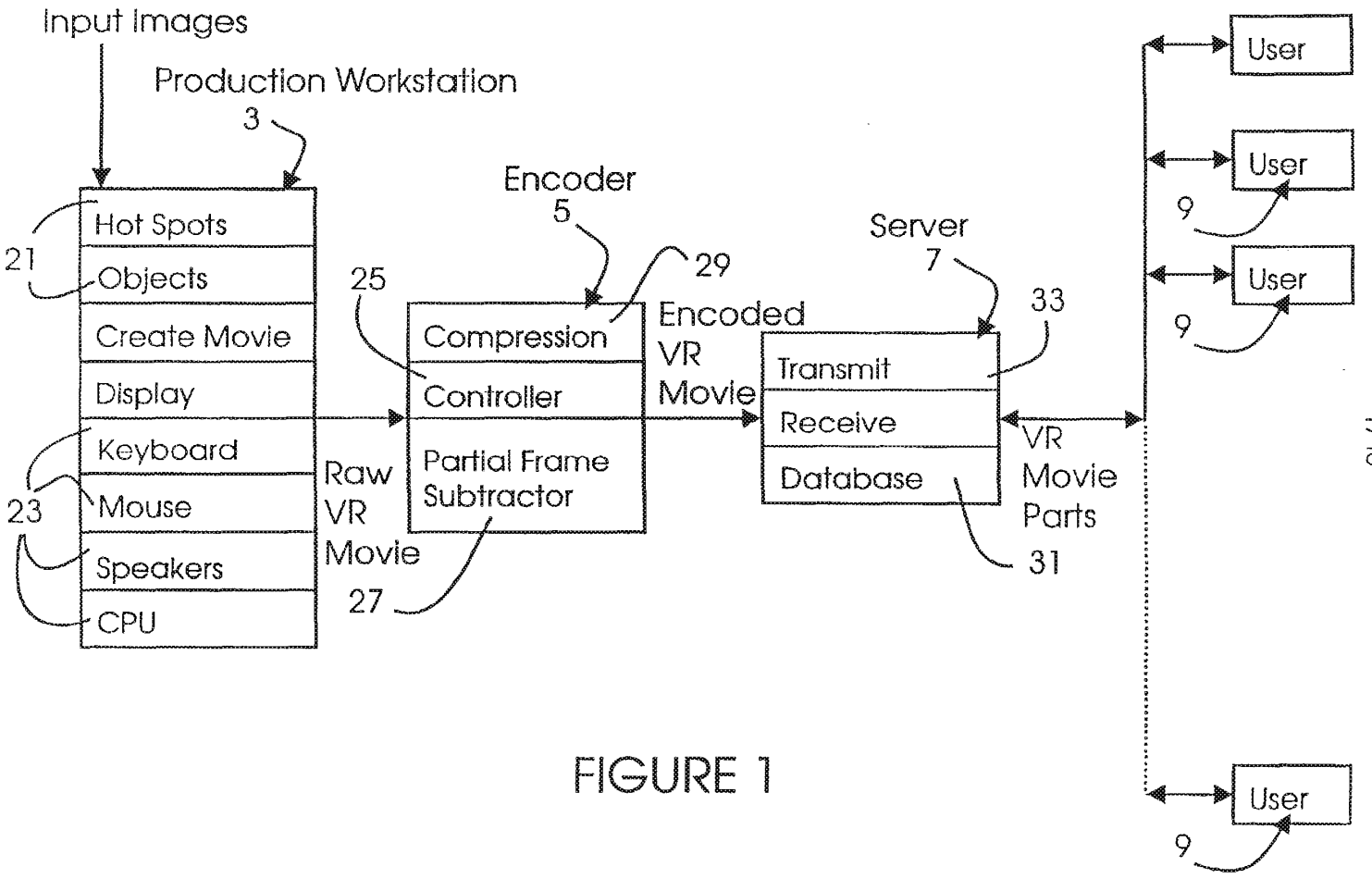


FIGURE 1

PCT/IL97/00055

Google Exhibit No. 1008
Page 226

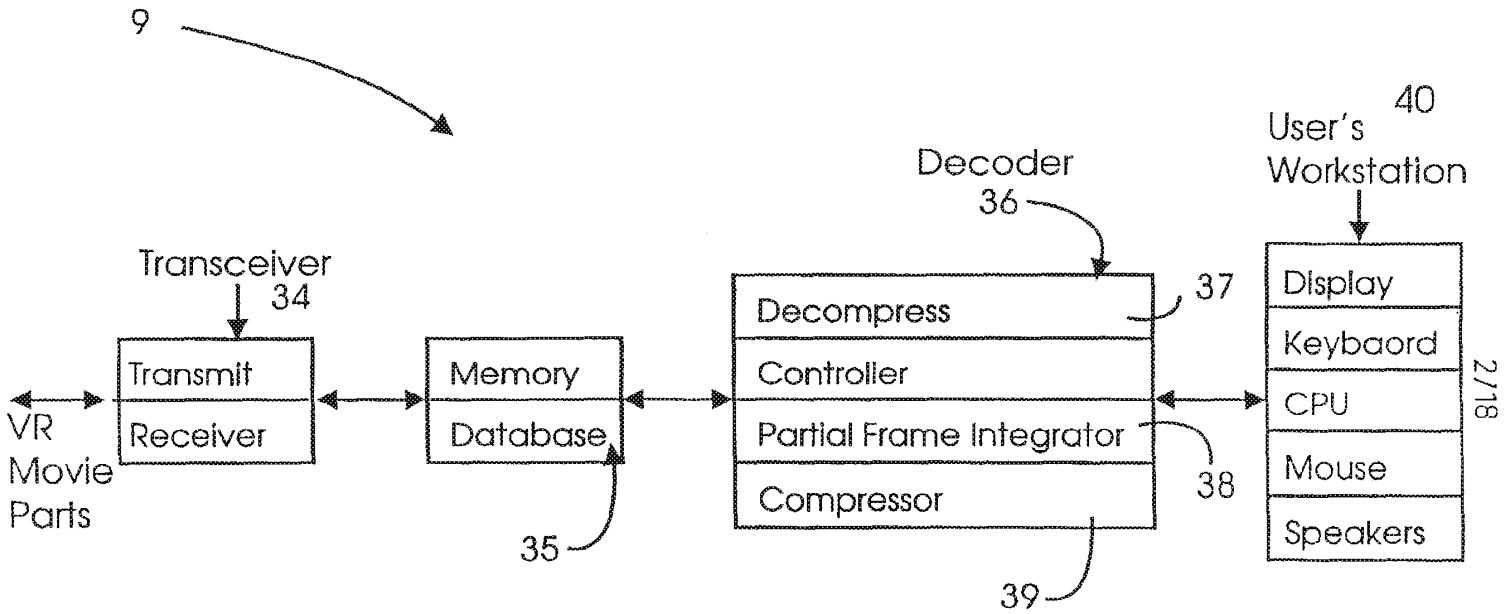


FIGURE 2

Google Exhibit No. 1008
Page 227

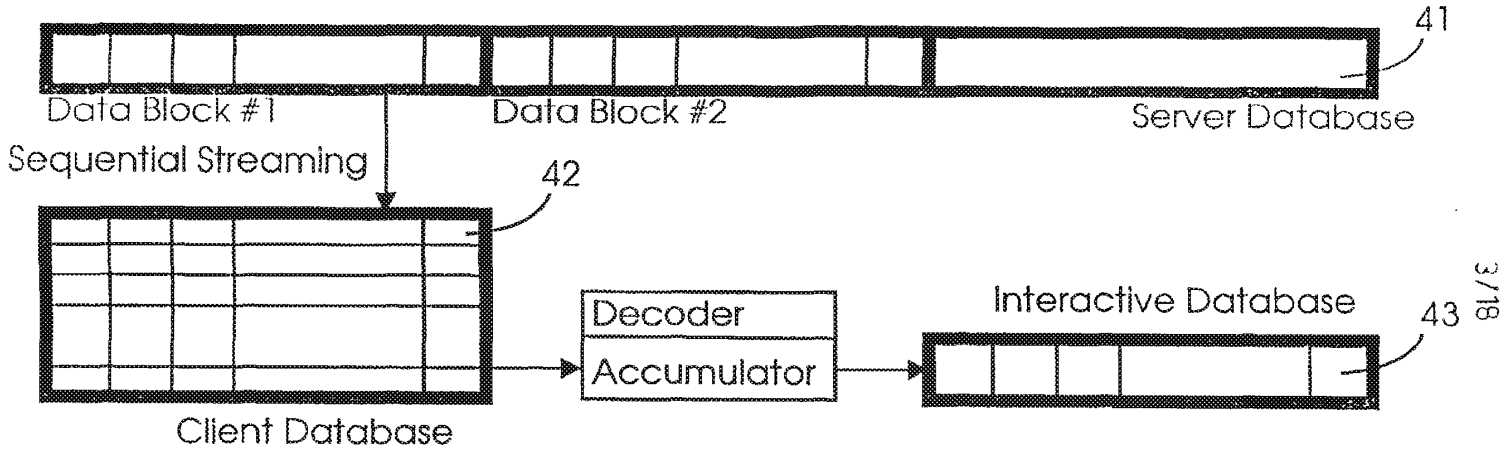

41
Data Block #1
Data Block #2
Server Database
Sequential Streaming
42
Decoder
Accumulator
Interactive Database
43
Client Database

FIGURE 3A

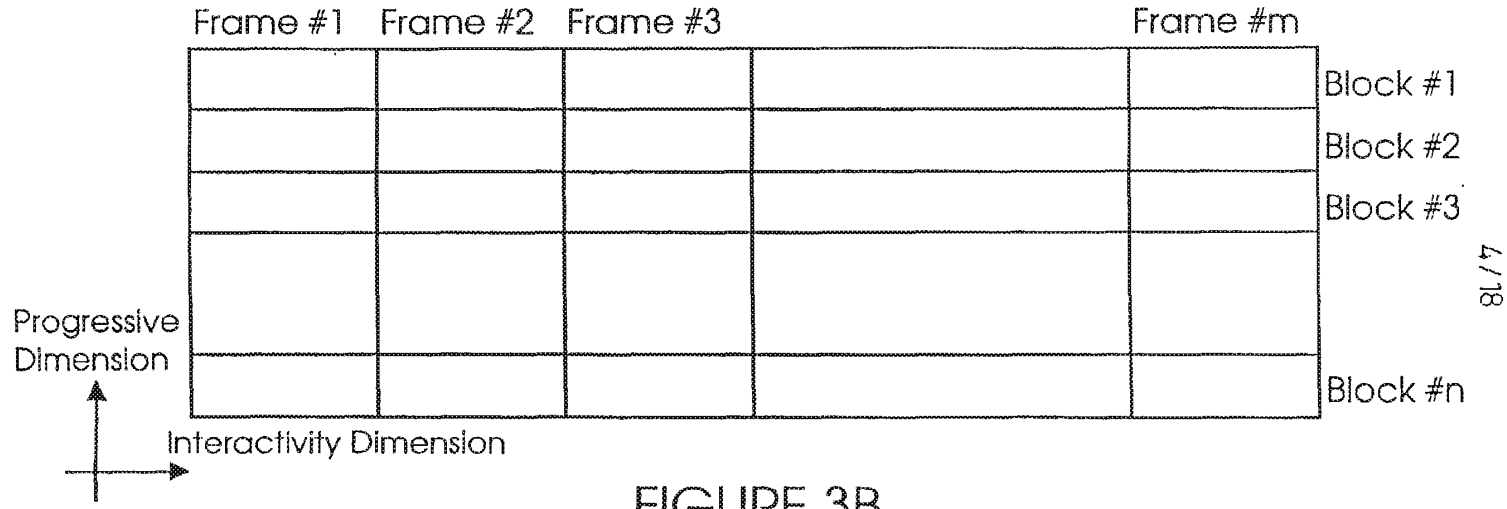

Frame #1
Frame #2
Frame #3
Frame #m
Block #1
Block #2
Block #3
Block #n
Progressive Dimension
Interactivity Dimension

FIGURE 3B

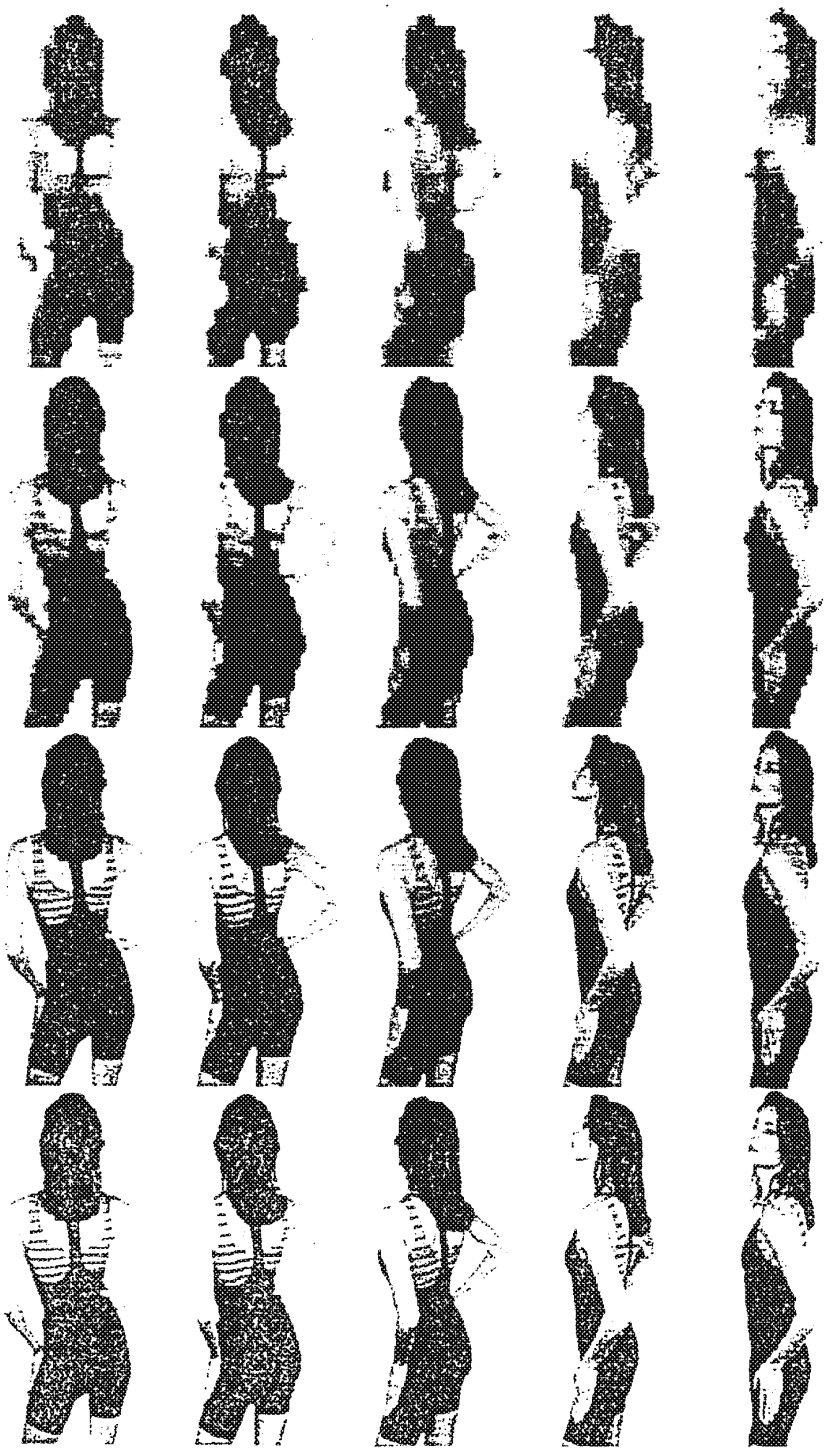


FIGURE 4

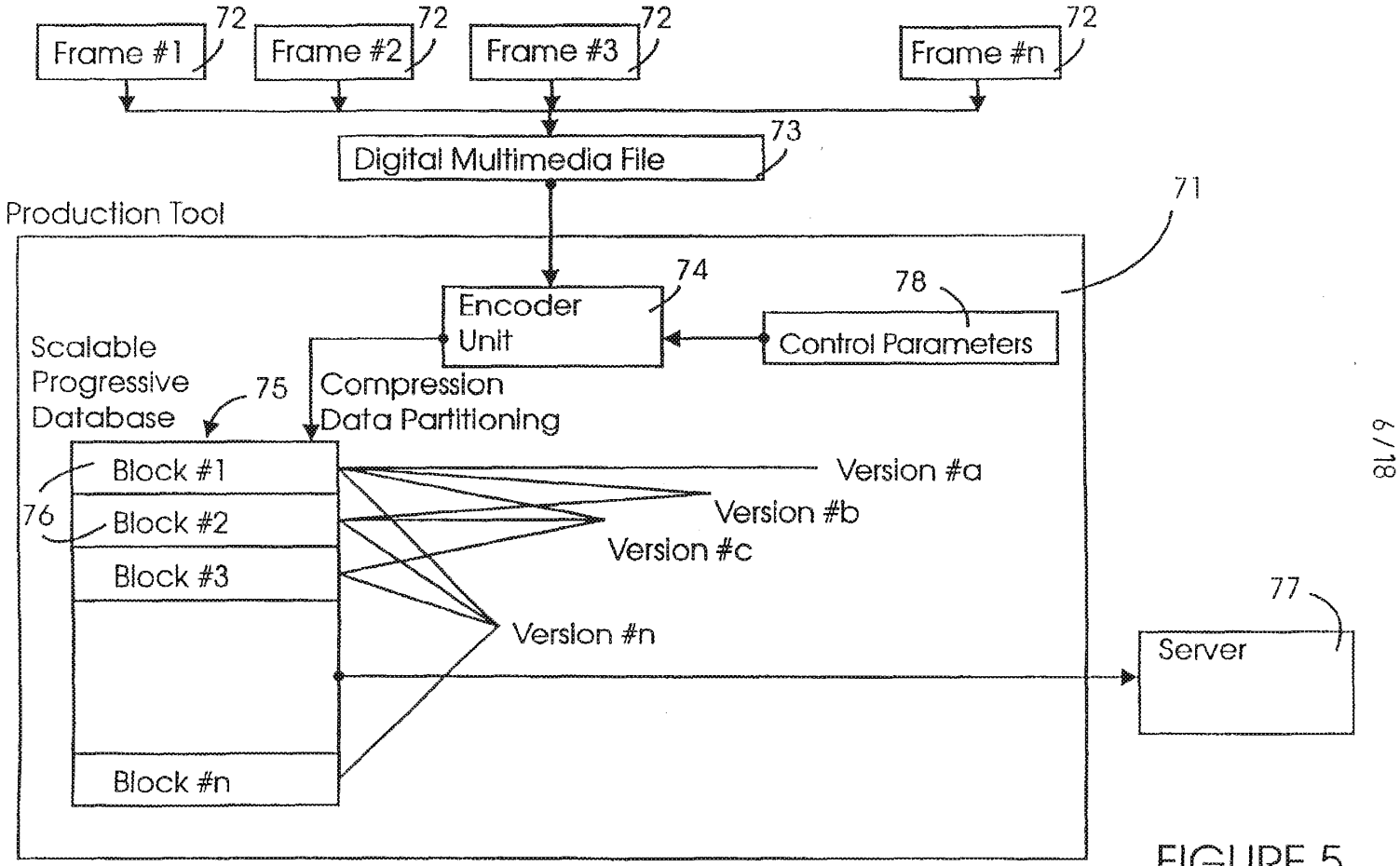

Frame #1
72
Frame #2
72
Frame #3
72
Frame #n
72
Digital Multimedia File
73
Production Tool
71
Encoder Unit
74
78
Control Parameters
Scalable Progressive Database
75
Compression Data Partitioning
Block #1
Block #2
Block #3
Block #n
76
Version #a
Version #b
Version #c
Version #n
77
Server

FIGURE 5

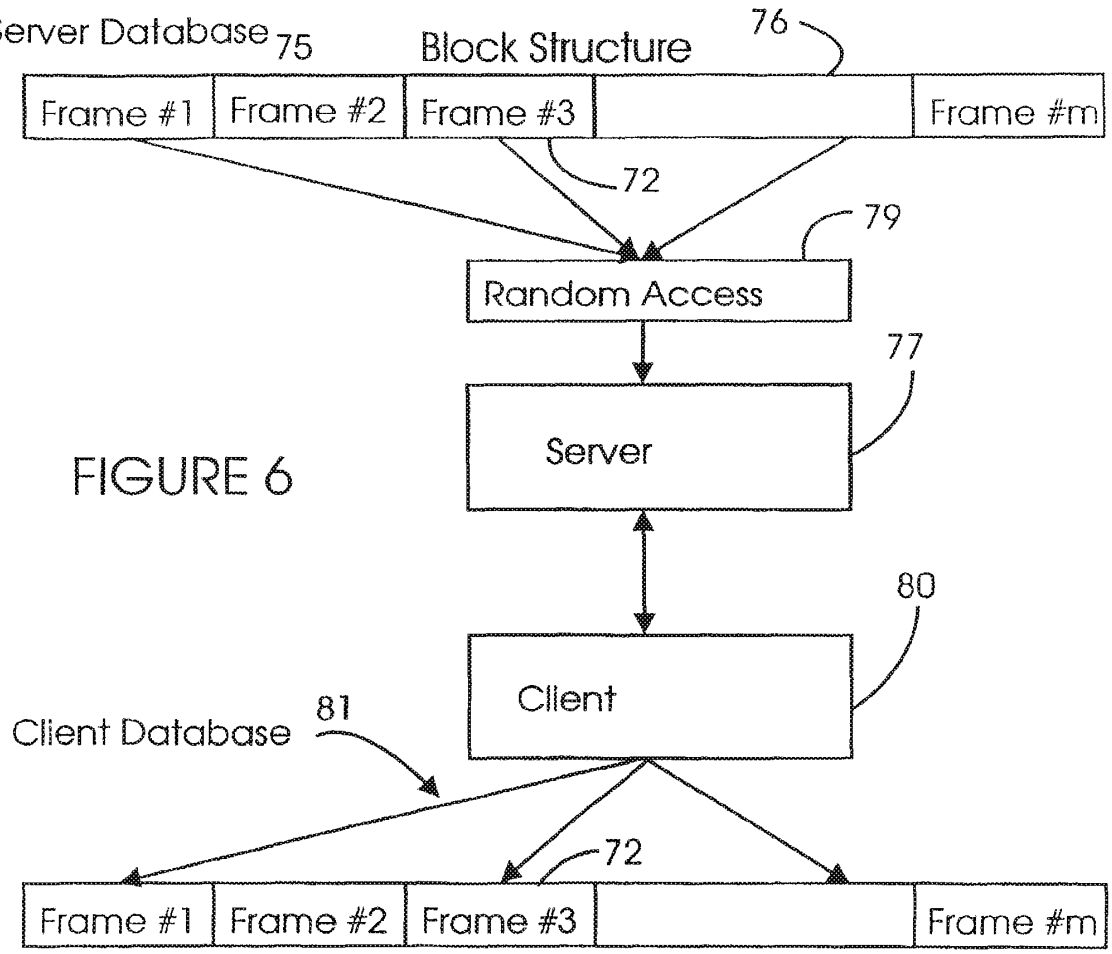

Server Database
75
Block Structure
76
Frame #1
Frame #2
Frame #3
Frame #m
72
79
Random Access
77
Server
80
Client
Client Database
81
72
Frame #1
Frame #2
Frame #3
Frame #m

FIGURE 6

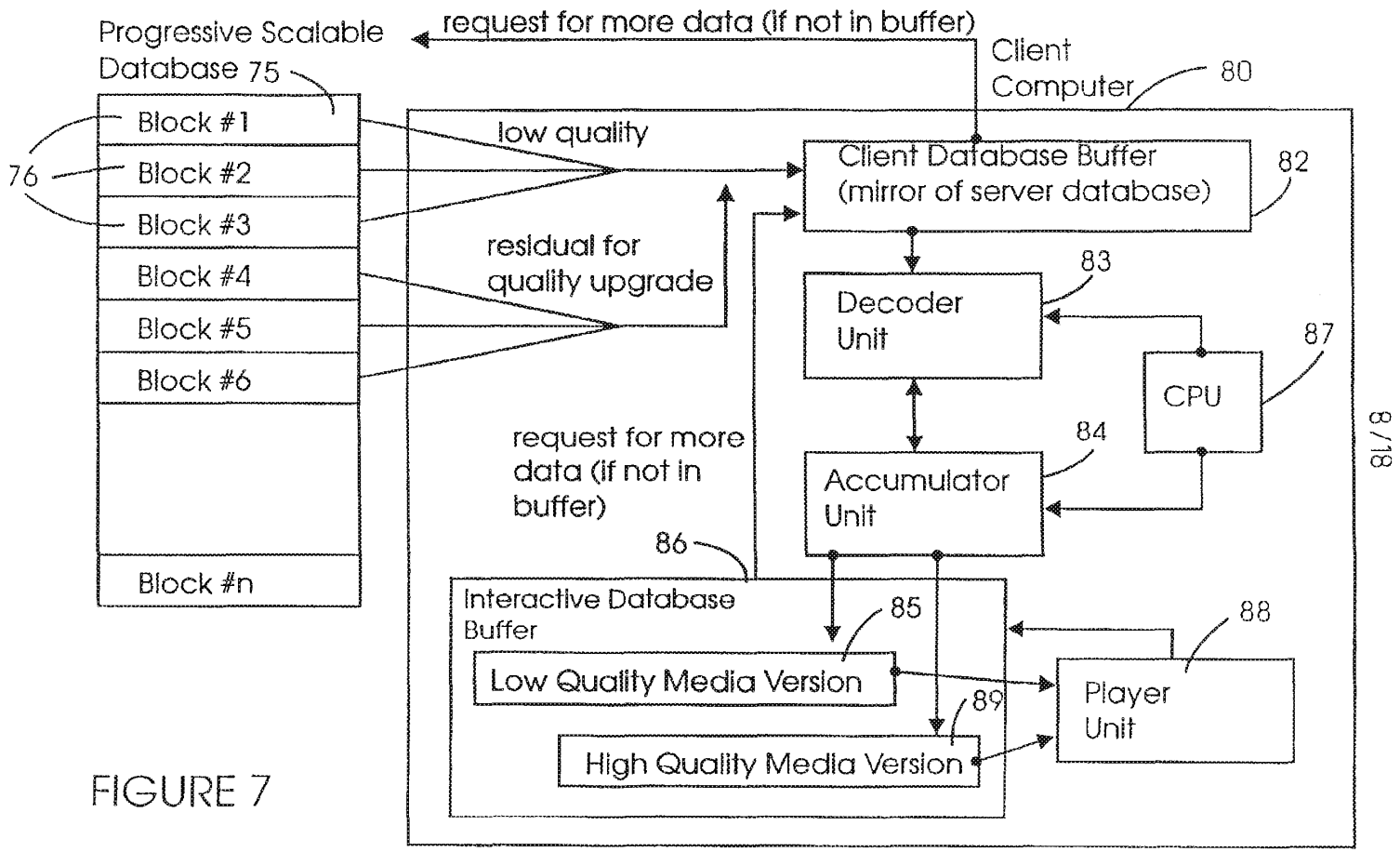


FIGURE 7

Google Exhibit No. 1008
Page 233

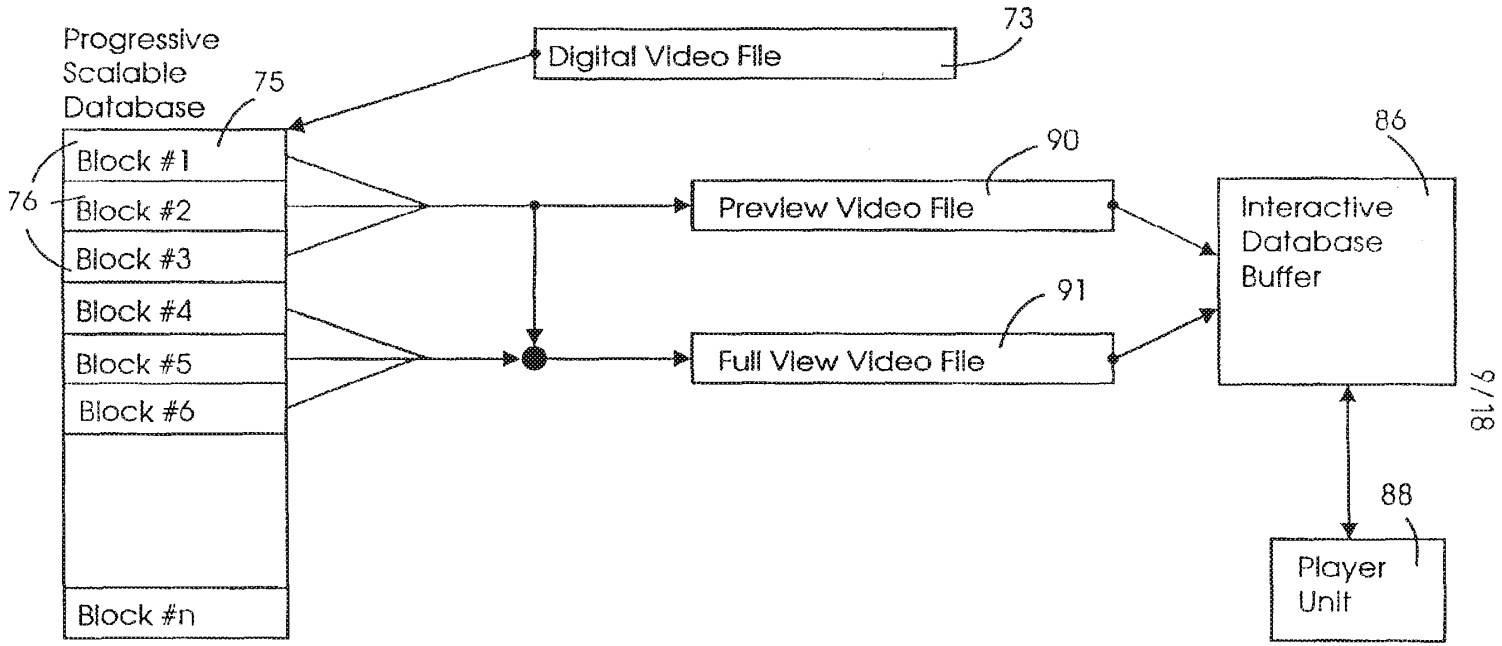


FIGURE 8

Google Exhibit No. 1008
Page 234

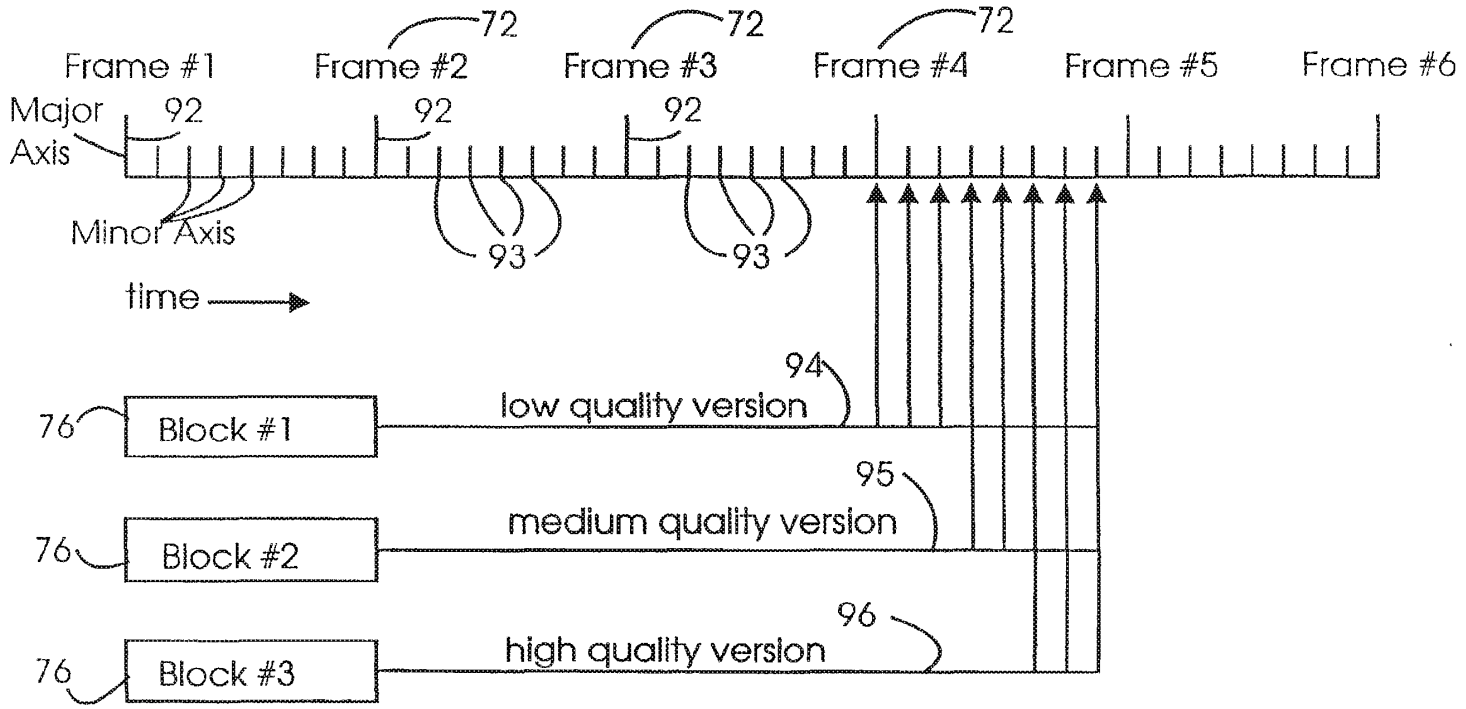

Frame #1
Frame #2
Frame #3
Frame #4
Frame #5
Frame #6
72
72
72
Major Axis
92
92
92
Minor Axis
93
93
time
94
low quality version
76
Block #1
95
medium quality version
76
Block #2
96
high quality version
76
Block #3

FIGURE 9

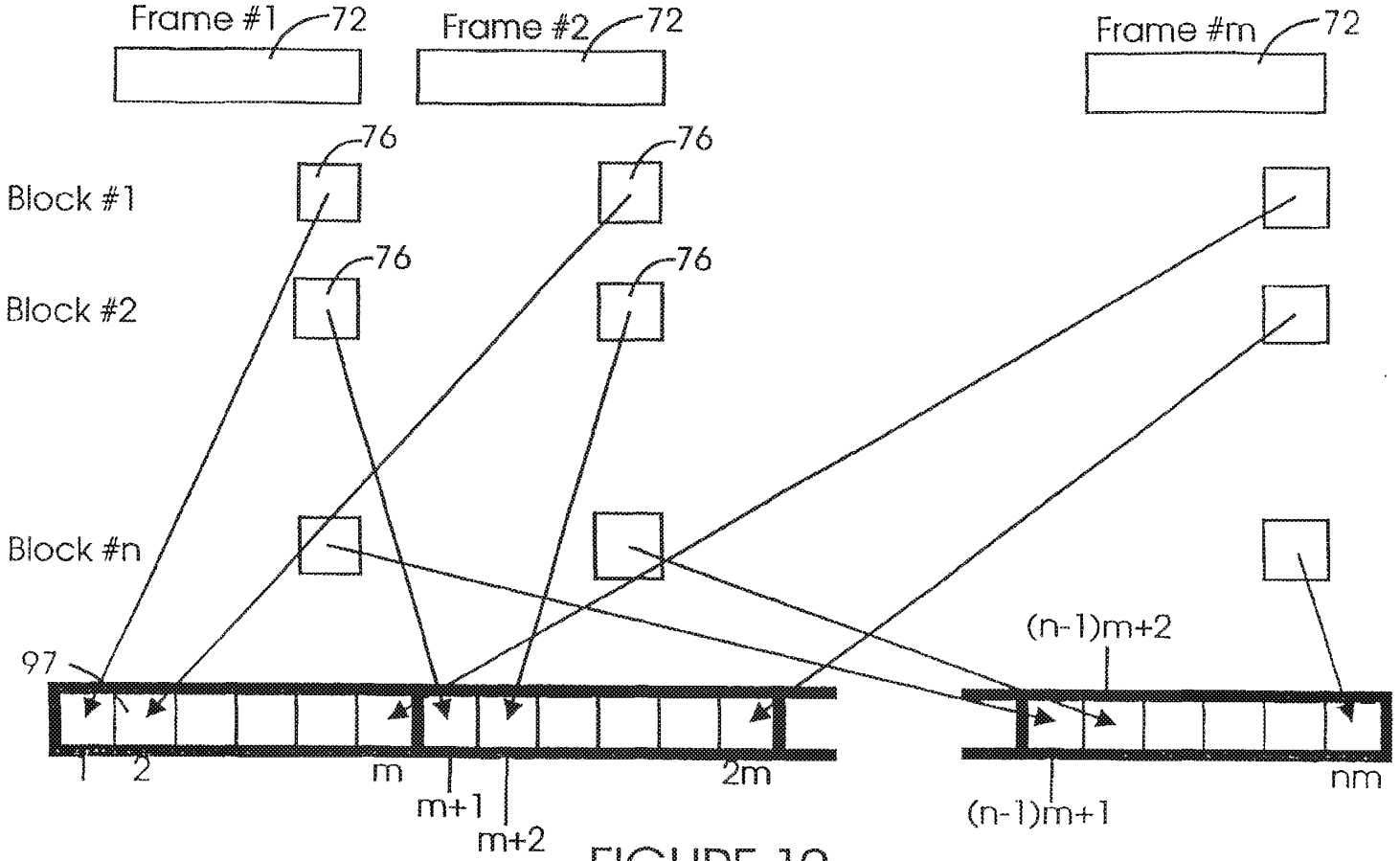


FIGURE 10

Google Exhibit No. 1008
Page 236

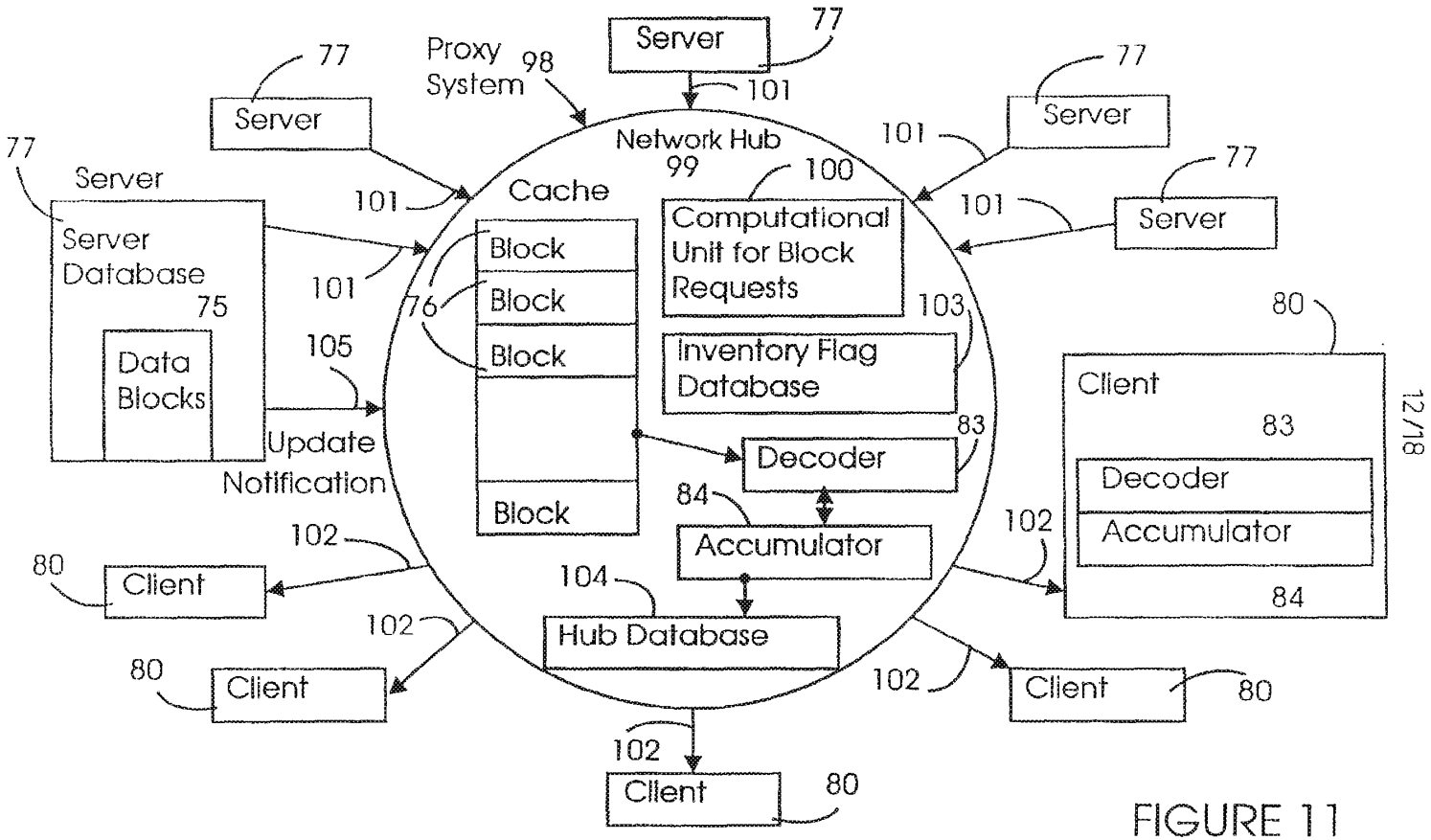

Server
77
Proxy System
98
Server
77
101
Server
77
101
Server
77
Server
77
Server Database
75
Data Blocks
101
105
Update Notification
Network Hub
99
Cache
Block
Block
Block
Block
76
100
Computational Unit for Block Requests
101
Server
77
101
103
Inventory Flag Database
80
Client
83
Decoder
Accumulator
84
12/18
83
Decoder
84
Accumulator
102
102
80
Client
104
Hub Database
102
80
Client
102
Client
80
102
Client
80

FIGURE 11

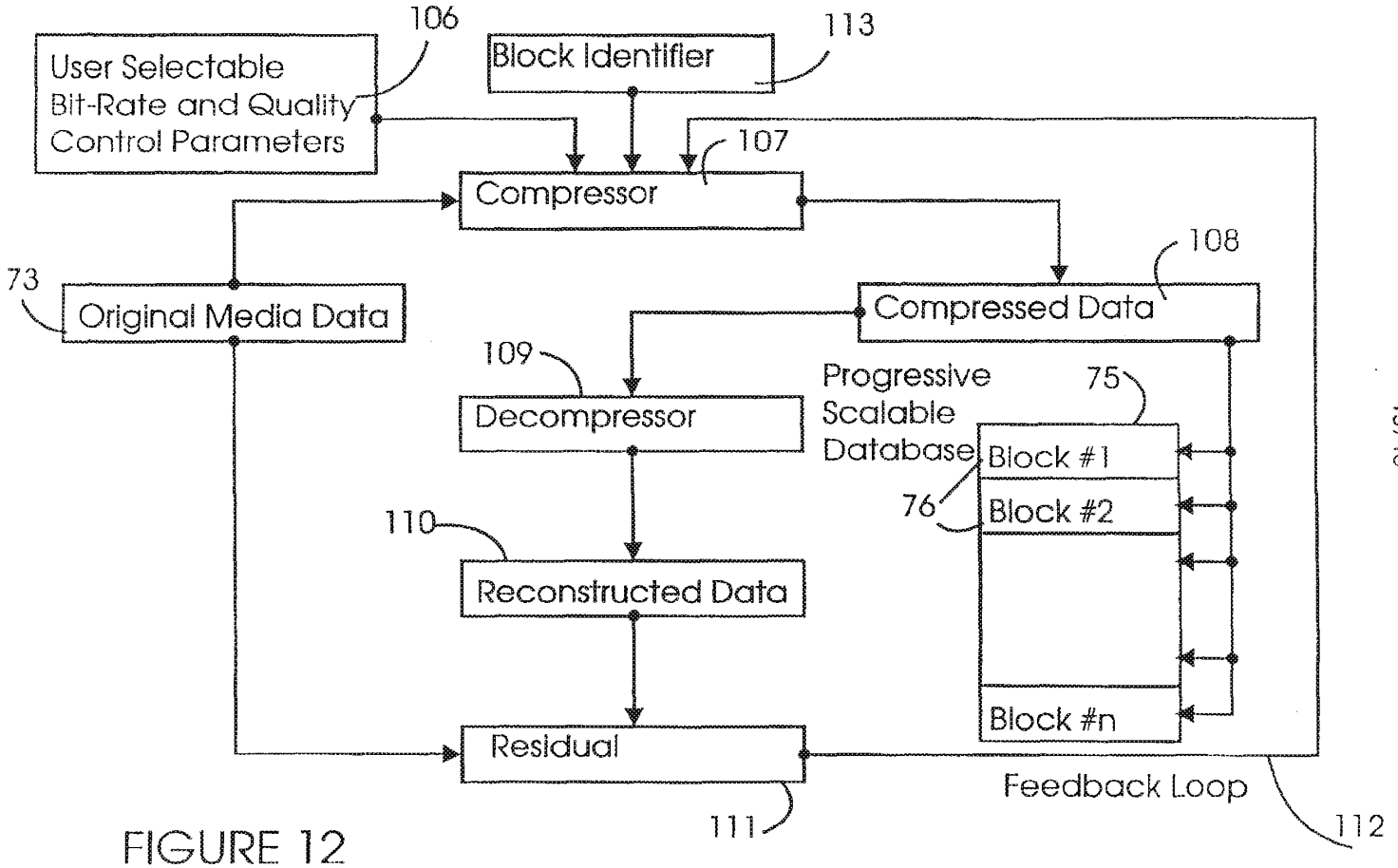

106
User Selectable
Bit-Rate and Quality
Control Parameters
113
Block Identifier
107
Compressor
73
Original Media Data
108
Compressed Data
109
Decompressor
Progressive
Scalable
Database
75
Block #1
76
Block #2
110
Reconstructed Data
Block #n
Residual
Feedback Loop
111
112

FIGURE 12

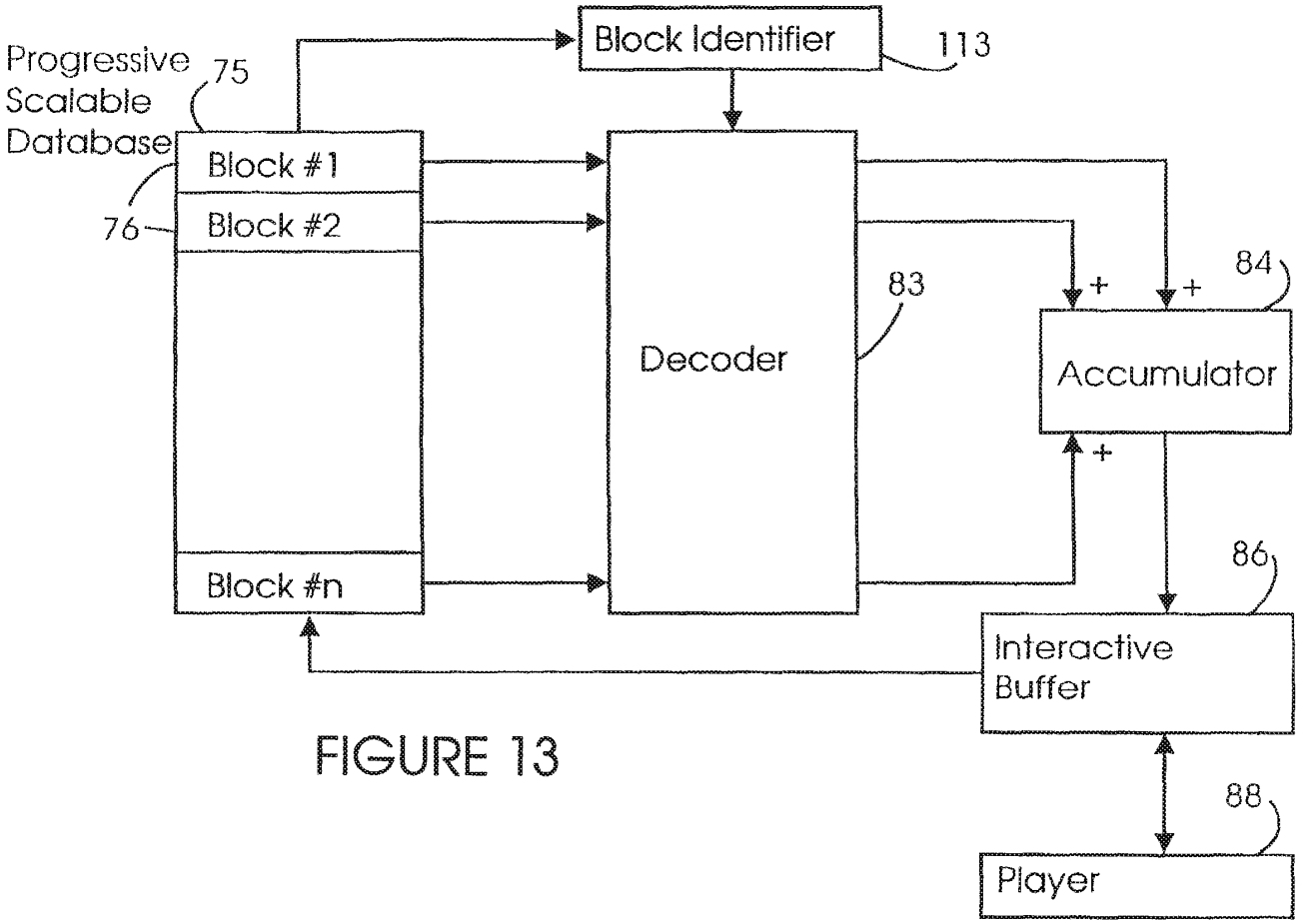

Block Identifier
113
Progressive
Scalable
Database
75
Block #1
Block #2
76
Block #n
Decoder
83
84
+
+
+
Accumulator
86
Interactive
Buffer
88
Player

FIGURE 13

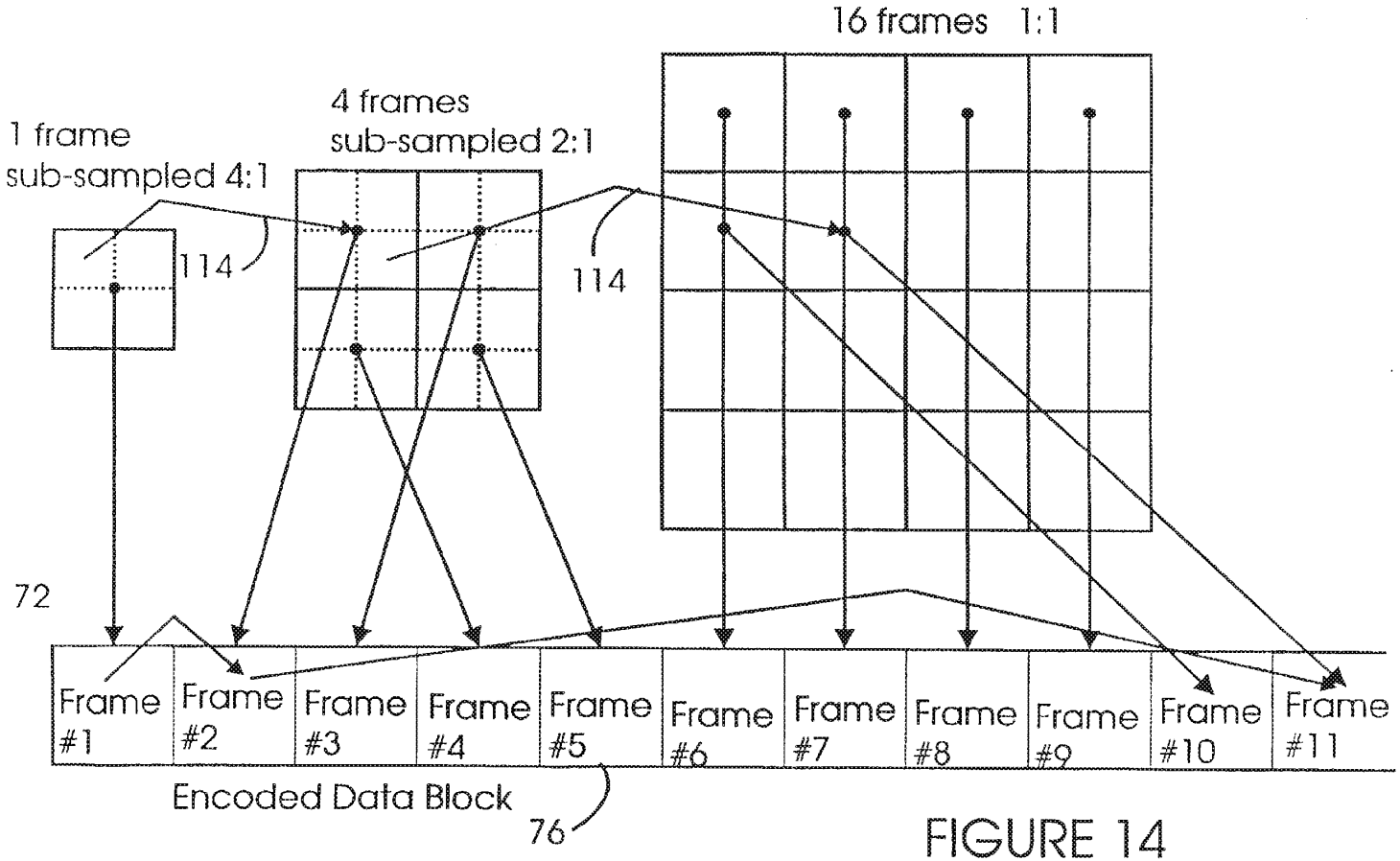

16 frames 1:1
4 frames
sub-sampled 2:1
1 frame
sub-sampled 4:1
114
114
72
Frame #1
Frame #2
Frame #3
Frame #4
Frame #5
Frame #6
Frame #7
Frame #8
Frame #9
Frame #10
Frame #11
Encoded Data Block
76

FIGURE 14

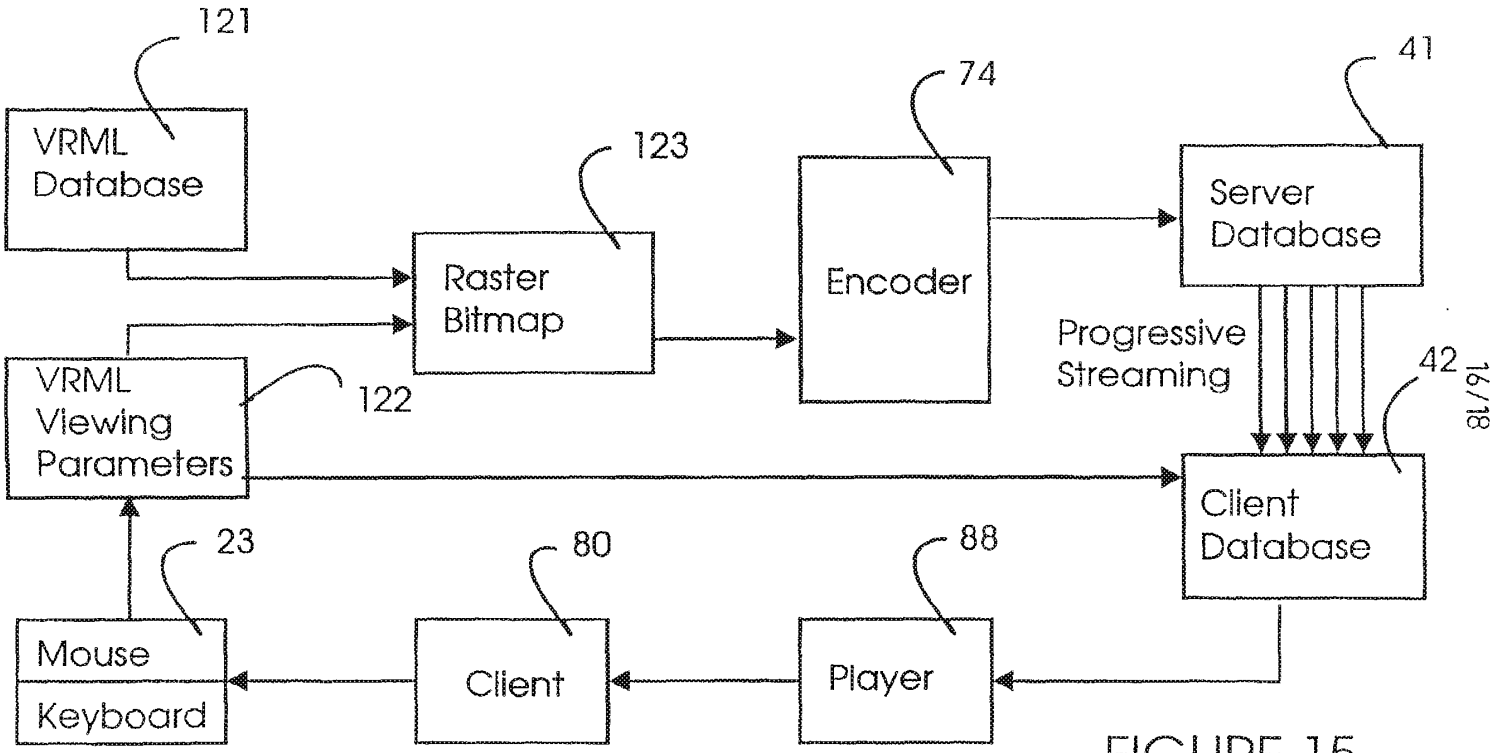

121
VRML Database
123
Raster Bitmap
74
Encoder
41
Server Database
Progressive Streaming
42
VRML Viewing Parameters
122
Client Database
23
Mouse
Keyboard
80
Client
88
Player

FIGURE 15

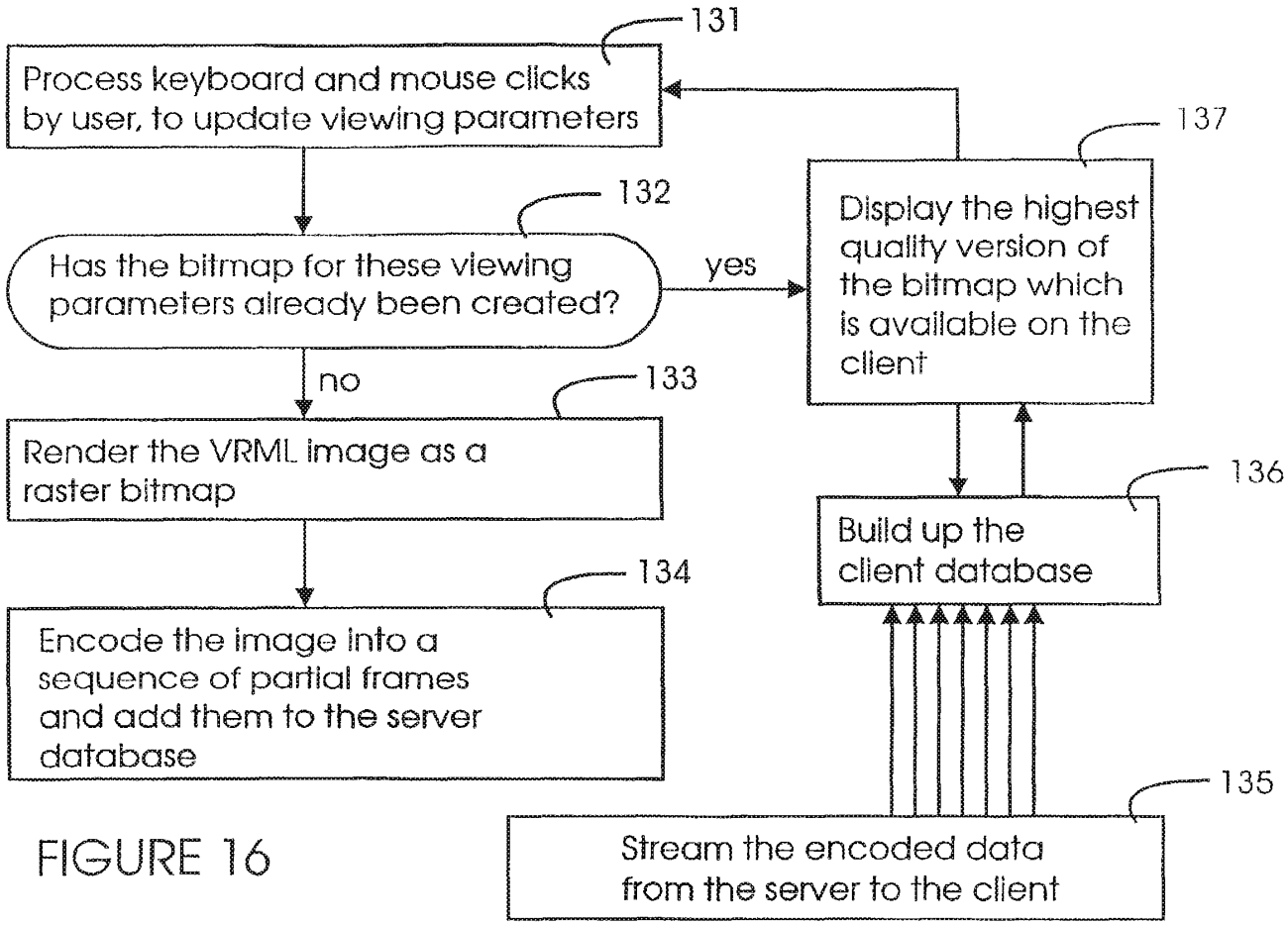

131
Process keyboard and mouse clicks by user, to update viewing parameters
132
Has the bitmap for these viewing parameters already been created?
yes
no
133
Render the VRML image as a raster bitmap
134
Encode the image into a sequence of partial frames and add them to the server database
137
Display the highest quality version of the bitmap which is available on the client
136
Build up the client database
135
Stream the encoded data from the server to the client

FIGURE 16



FIGURE 17

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IL97/00055

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :H04N 7/14
US CL :395/200.13; 348/7; 348/6; 455/4.2
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. : 395/200.13; 348/7; 348/6; 455/4.2

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, 5,414,455 A (HOOPER ET AL.) 09 May 1995, whole reference.	1-112
A,P	US, 5,528,281 A (GRADY ET AL.) 18 June 1996, col. 4-6.	1-112
A,P	US, 5,561,791 A (MENDELSON ET AL.) 01 October 1996, the whole reference.	1-112
A	US, 5,421,031 A (DE BEY) 30 May 1995, see col. 6-10.	1-112
A,P	US, 5,512,934 A (KOCHANSKI) 30 April 1996, col. 3-6.	1-112
A,P	US, 5,544,313 A (SHACHNAI ET AL.) 06 August 1996, the whole reference.	1-112

[X] 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 document 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
06 MAY 1997	26 JUN 1997
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 ZARNI MAUNG Telephone No. (703) 305-3900

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

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IL97/00055

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, 5,487,167 A (DINALLO ET AL.) 23 January 1996, the whole reference.	1-112

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

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 762 300 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
12.03.1997 Bulletin 1997/11

(51) Int. Cl.[6]: **G06F 17/30**, H04N 7/173, G06F 3/06

(21) Application number: **96114630.5**

(22) Date of filing: **12.09.1996**

(84) Designated Contracting States:
DE FR GB

(30) Priority: **12.09.1995 JP 234404/95**

(71) Applicant: **KABUSHIKI KAISHA TOSHIBA**
Kawasaki-shi, Kanagawa-ken 210 (JP)

(72) Inventors:
- **Yao, Hiroshi,**
  338 Toshiba-Shinkoyasu-Daiichi-ryo
  Yokohama-shi, Kanagawa-ken (JP)
- **Kanai, Tatsunori**
  Yokohama-shi, Kanagawa-ken (JP)
- **Kizu, Toshiki**
  Yokohama-shi, Kanagawa-ken (JP)
- **Maeda, Seiji**
  Fuchu-shi, Tokyo (JP)

(74) Representative: **Zangs, Rainer E., Dipl.-Ing. et al**
Hoffmann, Eitle & Partner
Arabellastrasse 4/VIII
81925 München (DE)

(54) Real time stream server for handling a plurality of real time stream data with different data rates

(57) A real time stream server capable of realizing a supply of a plurality of real time stream data with different data rates by a scheduling scheme using constant time-slot interval and transfer start timing period. A number of unit streams to be used and a block transfer time for each real time stream data are determined according to a data rate of each real time stream data. Each real time stream data is divided into a plurality of blocks, each block being in a size to be transferred within the block transfer time, and the blocks are sequentially distributed among the unit streams to be used. The blocks of each unit stream data are sequentially stored into a plurality of disk devices. In response to a request for each real time stream data from a client, the blocks constituting each real time stream data are read out from disk devices to a buffer memory, and each real time stream data is read out from a buffer memory and transferred to the client through a network, according to an appropriately scheduled transfer start timing for each unit stream.

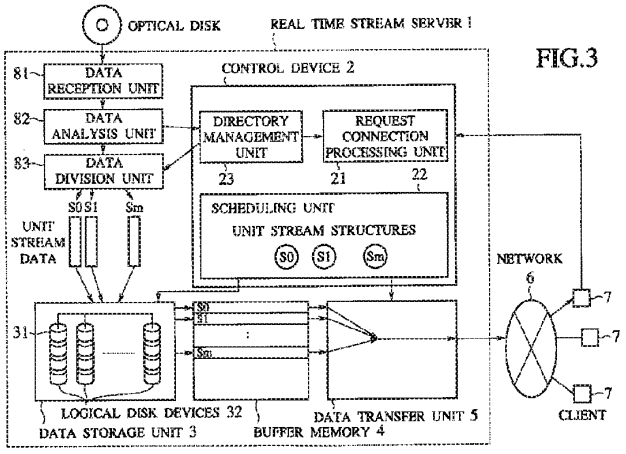


FIG.3

EP 0 762 300 A2

Printed by Rank Xerox (UK) Business Services
2.13.16/3.4

Google Exhibit No. 1008
Page 246

Description

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to a real time stream server for supplying a plurality of real time stream data in different data rates simultaneously to clients, and a method for operating the real time stream server.

Description of the Background Art

Data to be sequentially transferred in real time such as video data and audio data are generally called "real time stream data". For a real time stream server for handling such real time stream data, a necessary condition is that it should be able to transfer the real time stream data stored in disk devices to each client while guaranteeing a continuity in real time.

In order to satisfy this condition, in the prior art, the real time stream data are stored in disks by being divided into blocks of a size to be transferred in a prescribed period of time, and the server makes accesses to disks periodically for each stream. The read out blocks are stored in a buffer memory once, and periodically transferred to corresponding clients through a communication network.

A stream scheduling device is a device which realizes a management of timings for issuing disk access commands and data transfer commands in this operation.

In addition, the scheduling device establishes a new stream channel upon receiving a connection request from a user. In order to manage disk accesses and data transfer timings, the scheduling device provides time-slots as shown in Fig. 1 which are partitioned at constant time interval. One disk access is allocated to one time-slot, and one disk access reads out one block of the real time stream data. A period of disk access in one stream is constant, so that by allocating disk accesses of different streams to different time-slots, it becomes possible to share the same stored data among a plurality of streams.

In an example shown in Fig. 1, four time-slots are provided in one period. That is, three disk devices with a read multiplexing level 4 are provided, and blocks are striped over these three disk devices. Consequently, an access period for one disk device is $4 \times 3 = 12$ time-slots, and a maximum number of simultaneously connectable streams is 12.

As shown in Fig. 1, time-slots by which one stream makes accesses to the disk devices are distanced each other by the access period of four time-slots. For example, after a read block A1 (a block-1 of the real time stream data A) is read out, it is transferred to a client-0 and processed (e.g., reproduced) in real time, and then a block A2 is read out and transferred before the processing of a block A1 is finished. In this manner, the scheduling is made so that each stream does not influence the continuity of the other streams.

Fig. 1 shows an exemplary case of a scheme in which the disk access allocated to the time-slot is fixed, but there is also a scheme in which an allocation position of a disk access is made variable among time-slots within a tolerable jitter range, by noting the fact that a time-slot to which a disk access is to be allocated can be changed during a period since a buffer memory becomes available until a transfer start timing (see, Japanese Patent Application No. 7-57384 (1995). Here, a range of time-slots to which a disk access of one stream can be allocated is called tolerable jitter range of that stream.

Fig. 2 shows an exemplary tolerable jitter range. In a case where all the real time stream data to be handled have the same data rate, the scheduling becomes easier by a scheme utilizing the tolerable jitter range.

Now, in a case of handling real time stream data with a higher data rate, a block size becomes larger, and there can be a case in which a reading from a disk cannot be completed within one time-slot.

In order to avoid such a situation, if a time-slot interval and a block size are fixedly set in accordance with the real time stream data with the maximum data rate, there arises a problem in a case where real time stream data with a low data rate are to be supplied at the same time because it would become impossible to take a full advantage of a transfer capacity of the disk devices. This is due to the fact that, for the same period, the block size changes in proportion to the data rate.

On the other hand, when a time-slot interval is varied according to a block size, it becomes difficult to realize a flexible disk access scheduling in which orders of disk accesses can be interchanged. In addition, the buffer memory management becomes complicated in such a case, because there is a need to secure a continuous region in a size of each block size as the buffer memory.

Moreover, when a block size is fixed regardless of a data rate, a period of access to one disk device is going to be different for different streams, and for this reason, it becomes difficult to judge whether it is possible to connect a new stream while guaranteeing the continuity of the already connected streams.

Thus, in a case of supplying a plurality of real time stream data with different data rates, it has conventionally been difficult to realize both a scheme for making the disk access scheduling easier by fixing a period of access to one disk

Google Exhibit No. 1008
Page 247

device and a time-slot interval, and a scheme for taking a full advantage of a transfer capacity of disk devices by changing a number of data supply streams according to a data rate.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide a real time stream server and a method for operating a real time stream server, capable of realizing a supply of a plurality of real time stream data with different data rates by a scheduling scheme using constant time-slot interval and transfer start timing period, without wasting a transfer capacity of disk devices.

According to one aspect of the present invention there is provided a real time stream server, comprising: entering means for entering real time stream data to be stored in the real time stream server; determining means for determining a number of unit streams to be used and a block transfer time for the real time stream data, according to a data rate of the real time stream data; dividing means for dividing the real time stream data into a plurality of blocks, each block being in a size to be transferred within the block transfer time, and sequentially distributing the blocks among as many unit streams as the number of unit streams to be used; a plurality of disk devices for sequentially storing the blocks of each unit stream data; a buffer memory for temporarily storing the blocks read out from said plurality of disk devices; control means for reading out the blocks constituting the real time stream data from said plurality of disk devices to the buffer memory, and reading out the real time stream data from the buffer memory, according to a request for the real time stream data from a client; and transfer means for transferring the real time stream data read out from the buffer memory to the client through a network.

According to another aspect of the present invention there is provided a method for operating a real time stream server having a plurality of disk devices and a buffer memory, comprising the steps of: entering real time stream data into the real time stream server; determining a number of unit streams to be used and a block transfer time for the real time stream data, according to a data rate of the real time stream data; dividing the real time stream data into a plurality of blocks, each block being in a size to be transferred within the block transfer time, and sequentially distributing the blocks among as many unit streams as the number of unit streams to be used; sequentially storing the blocks of each unit stream data into said plurality of disk devices; controlling the real time stream server to read the blocks constituting the real time stream data from said plurality of disk devices to the buffer memory, and read out the real time stream data from the buffer memory, according to a request for the real time stream data from a client; and transferring the real time stream data read out from the buffer memory to the client through a network.

Other features and advantages of the present invention will become apparent from the following description taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a timing chart showing one conventional scheme for managing disk accesses and data transfer timings in a stream scheduling device.

Fig. 2 is a timing chart showing another conventional scheme for managing disk accesses and data transfer timings in a stream scheduling device.

Fig. 3 is a block diagram of one embodiment of a real time stream server according to the present invention.

Fig. 4 is a flow chart of an operation for storing real time stream data into memory devices from external in the real time stream server of Fig. 3.

Fig. 5 is a flow chart of an operation for supplying real time stream data according to a request from external in the real time stream server of Fig. 3.

Fig. 6 is a diagram showing exemplary transfer start timings for blocks read out from logical disk devices in a concrete example of the real time stream server of Fig. 3, in a case of using one unit stream.

Fig. 7 is a diagram showing one exemplary transfer start timings for blocks read out from logical disk devices in a concrete example of the real time stream server of Fig. 3, in a case of using two unit streams.

Fig. 8 is a diagram showing another exemplary transfer start timings for blocks read out from logical disk devices in a concrete example of the real time stream server of Fig. 3, in a case of using two unit streams.

Fig. 9 is a diagram showing exemplary transfer start timings for blocks read out from logical disk devices in a concrete example of the real time stream server of Fig. 3, in a case of using three unit stream.

Fig. 10 is a diagram showing exemplary transfer start timings for blocks read out from logical disk devices in a concrete example of the real time stream server of Fig. 3, in a case of using four unit stream.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

First, the main features of a real time stream server according to the present invention will be summarized briefly.

In the present invention, a unit stream having a reference data rate is defined. Then, in a case of supplying real time

Google Exhibit No. 1008
Page 248

stream data in excess of the data rate of the unit stream, as many unit streams as necessary for the required data rate are used to supply this real time stream data.

At first, the real time stream data is divided into unit streams, and stored into a plurality of disk devices. Namely, according to a data rate of entered real time stream data, a number of unit streams to be used and a block transfer time are determined. Then, the real time stream data is divided into a plurality of blocks, each in a size to be transferred with the block transfer time, and these blocks are sequentially allocated to unit stream data. Then, for each unit stream data, allocated blocks are stored by distributing them over a plurality of disk devices sequentially from a top block. Also, for each real time stream data, a number of unit streams to be used, an ID number of a disk device which stores the top block of each unit stream, and a recording position of each block on a disk device which stores that block are recorded.

When a request for this real time stream data so stored in the disk devices is received from a client, stream recourses for as many unit streams as a recorded number of unit streams to be used for the requested real time stream data are secured (reserved), and a transfer start timing of each unit stream is scheduled so that the blocks are transferred continuously in an order as in the original real time stream data. At this point, the transfer start timings of the unit streams are scheduled to be displaced one another by a block transfer time part.

In transferring each block, a corresponding block on the buffer memory is transferred within the block transfer time starting from the transfer start timing scheduled for each unit stream. Here, the transfer start timings of the unit streams are displaced one another by a block transfer time part, so that when the block transfer of a preceding unit stream is finished, the block transfer of a subsequent unit stream is started immediately, and the blocks of the original real time stream data are transferred while guaranteeing the continuity.

In this manner, according to the present invention, the real time stream data with various data rates can be handled by using an appropriate number of independent unit streams according to a data rate of each real time stream data, and therefore it becomes possible to realize the disk access scheduling based on a single scheme regarding a time-slot interval, a block size, a block transfer period, and a buffer memory management, without distinguishing different data rates.

Also, according to the present invention, the stream resources are divided into amounts corresponding to the unit streams, and allocated as much as necessary according to a data rate of each real time stream data to be supplied, so that it becomes possible to utilize the stream resources efficiently without wasting any resource.

Also, in a case of the real time stream server which has a capacity to supply Cmax sets of real time stream data with data rates not greater than R, where R is a reference data rate of a unit stream, a number of supplies Cm (m = 1, 2, $\cdots\cdots\cdots$) to be allocated to each real time stream data with a data rate in a data rate range greater than (m-1)×R and not greater than m×R can be set arbitrarily within a range of:

$$\sum_{m}(m \times Cm) \leq Cmax$$

where m is a number of unit streams to be used for each real time stream data.

Referring now to Fig. 3 to Fig.10, one embodiment of a real time stream server according to the present invention will be described in detail.

Fig. 3 shows a configuration of a real time stream server in this embodiment, Fig. 4 shows a flow chart of an operation for storing real time stream data into memory devices from external, and Fig. 5 shows a flow chart of an operation for supplying real time stream data according to a request from external.

As shown in Fig. 3, the real time stream server 1 of this embodiment comprises a control device 2, a data storage unit 3 connected with the control device 2, a buffer memory 4 connected with the data storage unit 3, a data transfer unit 5 connected with the control device 2 and the buffer memory 4 and connected to an external network 6, a data reception unit 81 for receiving externally entered real time stream data, a data analysis unit 82 connected with the data reception unit 81 and the control device 2, and a data division unit 83 connected with the control device 2 and the data storage unit 3.

The control device 2 has a request connection processing unit 21, a scheduling unit 22, and a directory management unit 23. The data storage unit 3 has a plurality of disk devices 31 for storing real time stream data in units of blocks.

First, with references to Fig. 3 and Fig. 4, one aspect of this real time stream server 1 regarding a phase for storing real time stream data acquired at the data reception unit 81 into the data storage unit 3 will be described.

Here, the "unit stream" in the real time stream server is defined as a stream which is scheduled by using a block transfer period T, a block size L, and a time-slot interval I which serve as references for the operation of the real time stream server.

The "unit data rate R" is defined as a maximum data rate that can be scheduled as a unit stream. Consequently, the following relationship holds among the block transfer period T, the block size L, and the unit data rate R.

Google Exhibit No. 1008
Page 249

$$L \leq R*T$$

A value of this unit data rate R is to be determined appropriately in view of optimizing the disk reading efficiency and the data transfer/storing efficiency.

Now, at a time of storing real time stream data in the real time stream server, the real time stream data to be stored is entered from the data reception unit 81 (step S11). Here, the data reception unit 81 may be in a form of reading data from an external memory device such as optical disk, floppy disk, CD-ROM, etc., or in a form of receiving data transferred through a network.

At the data analysis unit 82, a maximum data rate of the entered real time stream data is checked (step S12). To realize this checking, the maximum data rate may be obtained by analyzing a header of the real time stream data, or the maximum data rate known in advance may be entered by a human operator through an input device such as keyboard.

When the maximum data rate is obtained, a number of unit streams to be used for that real time stream data is determined (step S13). Here, a number of unit streams to be used is a numerical value indicating a capacity and resources of the real time stream server required for supplying that real time stream data, which is expressed in terms of a required number of the unit streams. Actually, a number of unit streams to be used can be determined as a minimum integer m which satisfies the following relationship.

$$\text{(Maximum data rate)} \leq m \times \text{(Unit data rate R)}$$

When the number of unit streams to be user is determined, the block transfer time which serves as a reference size for dividing the real time stream data into blocks is obtained according to the following formula (step S14).

$$\text{(Block transfer time)} = \text{(Block transfer period T)}/m$$

An amount of data to be transferred within this block transfer time obtained according to this formula never exceed the block size L of the unit stream.

The number of unit streams to be used and the block transfer time calculated at the data analysis unit 82 are then sent to the data division unit 83 along with the real time stream data.

At the data division unit 83, the real time stream data is divided into blocks, each in a size to be transferred within the block transfer time T/m (step S15). Then, these blocks are distributed among as many unit stream data as the number of unit streams to be used (step S16). Here, the distribution should preferably be done by using a fixed rule so that the schedules will not interfere among the unit stream data.

More specifically, for example, a set of blocks as a whole can be divided into m pieces of non-dense subsets Bj given by:

$$Bj = \{b(m \times k+j) \mid k = 0, 1, \cdots\cdots\cdots\} \quad (j = 0, \cdots\cdots\cdots, m-1)$$

and each such non-dense subset Bj can be identified as a unit stream data. For instance, in a case where m = 4 and there are sixteen blocks b0 to b15, the blocks are distributed among four unit stream data as follows.

$$B0 = \{b0, b4, b8, b12\}$$

$$B1 = \{b1, b5, b9, b13\}$$

$$B2 = \{b2, b6, b10, b14\}$$

$$B3 = \{b3, b7, b11, b15\}$$

These m pieces of unit stream data (B0 to Bm-1) are then sent to the data storage unit 3, and each unit stream data is stored into N sets of disk devices, in a similar manner as used for a unit stream data in a case of m = 1. More specifically, a top block bj of the unit stream data B0 is stored into a disk device of a disk device ID number Hj, while each block b(m×k+j) is stored into a disk device of a disk device ID number [(k+Hj) modulo N].

At this point, the disk device ID number Hj of the disk device which stores the top block of each unit stream, and a recording position of each block on a disk device which stores each block are recorded as a directory information in the directory management unit 23 within the control device 2 (step S18).

Here, a recording position on a disk may be specified by physical cylinder number, track number, and sector number, or by a logical number that can identify a specific position.

Also, a manner of sending m pieces of unit stream data obtained at the data division unit 83 to the data storage unit

Google Exhibit No. 1008
Page 250

3 can be in a form of once storing all the unit stream data on another memory device provided separately from the data storage unit 3 of the real time stream server and then sending each unit stream to the data storage unit 3 one by one, or in a form of sending all m pieces of unit stream data to the data storage unit 3 in parallel.

Next, with references to Fig. 3 and Fig. 5, another aspect of this real time stream server 1 regarding a phase for transferring the real time stream data to a client 7 upon receiving a request for supply of the real time stream data will be described.

Note that, generally speaking, there are two types of a scheduling scheme for allocating a time-slot to a disk access, including a fixed allocation type and a variable allocation type, and the present invention is applicable to either type of the scheduling scheme. In the following, it is only assumed that the scheduling is to be made in such a manner that the continuity of streams is guaranteed, without limiting the scheduling scheme to either one of these two types.

When a request for supply of the real time stream data is received from a client 7 through the network 6 (step S21), the request connection processing unit 21 first obtains a directory information for the requested real time stream data from the directory management unit 23 (step S22).

Then, the number m of unit streams to be used is obtained from the directory information (step S23), and as many stream structures in the scheduling unit 22 as necessary for holding information required in managing m pieces of unit streams are secured (reserved), while a necessary amount of regions in the buffer memory 4 are secured (reserved) (step S24).

Next, the scheduling unit 22 carries out the scheduling including a selection of transfer start timings for the unit streams S0 to Sm-1 to be used (step S25). Here, by the real time stream data storing procedure described above, m pieces of blocks b(m×k+j) (j = 0, ・・・・・・・・・, m-1) which are continuous in the original real time stream data are sequentially distributed among the unit streams S0 to Sm-1. Consequently, in order to carry out the transfer of these blocks continuously, the transfer start timings of the unit streams S0 to Sm-1 are displaced one another by the block transfer time T/m part.

Here however it is necessary for each one of the unit streams S0 to Sm-1 to select a time-slot for carrying out the disk access, so that it becomes possible to read out the respective top block from the disk device 31 which stores that top block, before the selected transfer start timing, without affecting the continuity of the other already connected unit streams. Note that the ID number of the disk device 31 which stores the top block of each unit stream can be obtained from the directory information obtained at the step S22. This time-slot selection operation will be described in detail below. When this condition is not satisfied, it is necessary to select different transfer start timings anew.

Each one of the unit streams S0 to Sm-1 so connected is then scheduled as an independent unit stream for which the block transfer time for one block is T/m, that is, scheduled according to the block transfer period T, the block size L, the time-slot interval l, and the block transfer time T/m. Then, according to this scheduling, each block is read out from the buffer memory 4 from the recording position of each block on a disk device which is indicated by the directory information, and supplied to the client 7 from the data transfer unit 5 (step S26).

According to this embodiment of the present invention, the real time stream data with a data rate not greater than m×R are handled as m pieces of independent unit streams, so that it becomes possible to realize the disk access scheduling based on a single scheme regarding a time-slot interval, a block size, a block transfer period, and a buffer memory management.

Also, according to this embodiment of the present invention, in a case of the real time stream server which has a capacity to supply Cmax sets of real time stream data with data rates not greater than R, a number of supplies Cm (m = 1, 2, ・・・・・・・・・) to be allocated to each real time stream data with a data rate in a data rate range greater than (m-1)×R and not greater than m×R can be set arbitrarily within a range of:

$$\sum_{m}(m \times Cm) \leq Cmax$$

where m is a number of unit streams to be used for each real time stream data, as should be apparent from the above description.

Now, with references to Fig. 6 to Fig. 10, concrete examples of the real time stream server of the present invention will be described.

First, an exemplary case of the real time stream server which stores and supplies the real time stream data by using a unit stream with a unit data rate R = 1.6 (Mbps) will be described.

When the block transfer time for one block is selected to be 2400 (ms), the maximum size of one block is 500 (kB). Assuming the the transfer rate of the disk device to be used is 20 (Mbps), there is a need to set the time-slot interval to be not less than 200 (ms) in order to read out one block by one time-slot.

In order to raise the transfer rate higher, one block can be divided and stored over a plurality of disk devices 31. For example, considering four disk devices 31 as one set, one block can be divided into four and stored into four disk

Google Exhibit No. 1008
Page 251

devices 31 respectively, and at a time of reading, all of these four disk devices 31 of the same set can be accessed simultaneously. In this manner, the effective maximum block size on a disk becomes 125 (kB), and the time-slot interval of not less than 50 (ms) becomes necessary. In the following, a set of these four disk devices 31 is regarded as one logical disk device 32.

When the time-slot interval is set to be 50 (ms) by regarding a set of four disk devices as one logical disk device as described above, a number of unit streams that can be supplied by one logical disk device is 2400/50 = 48 pieces. In addition, when eight of such logical disk devices are provided, it becomes possible to supply 48 × 8 = 384 pieces of the unit streams.

In a case of handling the unit streams in such a real time stream server, the block bk is to be stored in the logical disk device with a logical disk device ID number [(k+H) modulo 8], where H is a logical disk device ID number of the logical disk device which stores the top block b0.

In order to supply these unit streams in accordance with the request from the client 7, the disk devices storing the block to be transferred next are accessed at a period of 2400 (ms), and the block of not greater than 500 (kB) in total is read out to the buffer memory 4, 125 (kB) from each one of four disk devices corresponding to one logical disk device, within the time-slot interval 50 (ms). This block is then transferred to the client 7 at the period of 2400 (ms). in the transfer time of 2400 (ms).

Fig. 6 shows exemplary transfer start timings for blocks read out from the logical disk devices storing blocks of one unit stream in this case. In Fig. 6, the top block is stored in the logical disk device-0 (disk-0). Note however that there is no need for a disk device to store the top block to be the logical disk device-0, and it is also possible to store the top block of different real time stream data in different logical disk devices.

Next, in the real time stream server of the concrete example described above, a case of handling the real time stream data with a data rate of 3 (Mbps) will be considered. Here, the minimum integer m which satisfies $3 \leq m \times R = m \times 1.6$ (Mbps) is 2, so that it is possible to store and supply the real time stream data with a date rate of 3 (Mbps) by using 2 pieces of unit streams with the block transfer time for one block equal to 1200 (ms). In order to store the real time stream data with a date rate of 3 (Mbps), this data is divided into blocks, each in a size to be transferred in 1200 (ms), and a set of these blocks are divided into m = 2 subsets:

$$B0 = \{b(2k) \mid k = 0, 1, \cdots\cdots\cdots\}$$

$$B1 = \{b(2k+1) \mid k = 0, 1, \cdots\cdots\cdots\}$$

Then, the block b(2k+j) is stored into the logical disk device with a logical disk device ID number [(k+Hj) modulo 8]. In other words, for B0, the top block b0 is stored into the logical disk device with a logical disk device ID number H0, the next block b2 is stored into the logical disk device with a logical disk device ID number [(H0+1) modulo 8], the next block b4 is stored into the logical disk device with a logical disk device ID number [(H0+2) modulo 8], and so on.

In order to supply this stream in response to the request from the client 7, two unit streams S0 and S1 for supplying B0 and B1 respectively are prepared.

Then, the reference time-slots for the unit stream S0 and the unit stream S1 are selected such that the transfer start timings of the block b0 and the block b1 are displaced by 2400/2 = 1200 (ms). Then, the disk access is allocated to the time-slot in a similar manner as used for a unit stream in a case of m = 1, but here it must be possible to realize the disk access allocation for the unit stream S0 and the disk access allocation for the unit stream S1 simultaneously. If the simultaneous disk access allocations are impossible, the reference time-slots of the unit streams S0 and S1 are to be selected anew.

When the reference time-slots of the unit streams S0 and S1 are determined in this manner, it suffices to schedule the unit streams S0 and S1 as independent unit streams. In other words, for each one of the unit stream S0 and S1, a block to be transferred next by using the time-slot to which the disk access is allocated is read out from the logical disk device which stores this block to the buffer memory 4, and then this block is transferred to the client 7 at the period of 2400 (ms), in the transfer time of 1200 (ms). Here, only the transfer time differs from a case of a usual unit stream. A size of one block is approximately 460 (kB), which is less than the maximum block size of 500 (kB) for a usual unit stream, so that the time-slot interval and the buffer memory management can be exactly the same as in a case of a usual unit stream.

Fig. 7 shows exemplary transfer start timings for blocks read out from the logical disk devices storing blocks of the unit streams S0 and S1 in this case. In Fig. 7, the top blocks of the unit streams S0 and S1 are stored in the logical disk devices with the logical disk device ID numbers H0 = 0 and H1 = 4. Note however that any other combination of the logical disk devices to store the top blocks may be used. Fig. 8 shows another exemplary transfer start timings in this case, where H0 = H1 = 0 , so that the block b0 and the block b1 are stored in the same logical disk device in this case.

Next, in the real time stream server of the concrete example described above, a case of handling the real time stream data with a data rate of 4.5 (Mbps) will be considered. Here, the minimum integer m which satisfies $4.5 \leq m \times R = m \times 1.6$ (Mbps) is 3, so that it is possible to store and supply the real time stream data with a date rate

Google Exhibit No. 1008
Page 252

of 4.5 (Mbps) by using 3 pieces of unit streams with the block transfer time for one block equal to 800 (ms).

In order to store the real time stream data with a date rate of 4.5 (Mbps), this data is divided into blocks, each in a size to be transferred in 800 (ms), and a set of these blocks are divided into m = 3 subsets:

$$B0 = \{b(3k) \mid k = 0, 1, \cdots\cdots\cdots\}$$

$$B1 = \{b(3k+1) \mid k = 0, 1, \cdots\cdots\cdots\}$$

$$B2 = \{b(3k+2) \mid k = 0, 1, \cdots\cdots\cdots\}$$

Then, the block b(3k+j) is stored into the logical disk device with a logical disk device ID number [(k+Hj) modulo 8].

In order to supply this stream in response to the request from the client 7, three unit streams S0, S1 and S2 for supplying B0, B1 and B2 respectively are prepared.

Then, the reference time-slots for the unit streams S0, S1 and S2 are selected such that the transfer start timings of the respective blocks are displaced by 2400/3 = 800 (ms).

Fig. 9 shows exemplary transfer start timings for blocks read out from the logical disk devices storing blocks of the unit streams S0, S1 and S2, for an exemplary case of setting H0 = 0, H1 = 3, and H2 = 6.

Similarly, in a case of handling the real time stream data with a data rate of 6 (Mbps), the minimum integer m which satisfies $6 \leq m \times R = m \times 1.6$ (Mbps) is 4, so that it is possible to store and supply the real time stream data with a date rate of 6 (Mbps) by using 4 pieces of unit streams with the block transfer time for one block equal to 600 (ms).

Fig. 10 shows exemplary transfer start timings for blocks read out from the logical disk devices storing blocks of the unit streams S0, S1, S2 and S3, for an exemplary case of setting H0 = 0, H1 = 2, H2 = 4, and H3 = 6.

Note that, in the real time stream server of the concrete example described above, a number of unit streams that can be supplied is set as 384, but it is possible to change a number of unit streams to be used arbitrarily according to the request from the client 7 as long as a total number of unit streams to be used is within 384. For example, it is possible to supply 96 sets of streams with a data rate of 6 (Mbps) (corresponding to a total number of unit streams equal to 384) alone, or 48 sets of streams with a data rate of 6 (Mbps) (corresponding to a number of unit streams equal to 192) and 192 sets of streams with a data rate of 1.5 (Mbps) (corresponding to a number of unit streams equal to 192) together.

Next, a manner of selecting the time-slots at the connection request processing unit 21 in the real time stream server 1 of Fig. 3 will be described.

In order to supply the real time stream server with a data rate of m×R, it is necessary to allocate m pieces of disk accesses to the time-slots so that the m pieces of unit streams can be supplied continuously as described above.

Now, as already mentioned above, generally speaking, there are two types of a scheduling scheme for actually allocating a time-slot to a disk access, including a fixed allocation type and a variable allocation type.

The former is the scheduling scheme in which the disk access is fixed to the time-slot which is positioned a certain period of time away relatively from the reference time-slot as shown in Fig. 1. In other words, it is the scheduling scheme which fixes a positional relationship between the transfer timing and the disk access timing on the time-slots.

The latter is the scheduling scheme in which the allocation position of the disk access is made variable among the time-slots within the tolerable jitter range as shown in Fig. 2, by noting the fact that a time-slot to which a disk access is to be allocated can be changed during a period since a buffer memory becomes available until a transfer start timing.

Note here that, in Fig. 2, J indicates the maximum jitter number which is determined according to the following formula.

$$J \leq BM - D - T - 1$$

where B is a ratio of a size of a buffer memory that can be used by one stream and a size of one block of the real time stream data, M is a time (a number of slots) for reproducing one block at a client, T is a time (a number of slots) for transferring one block to a client, and D is an estimated maximum delay time (a number of slots) in a case where the disk access end timing extends beyond the end timing of the allocated time-slot. In an example shown in Fig. 2, a time required for transferring one block to the client side and a time required for reproducing one block at the client side are assumed to be equal to each other, as a time equivalent to 4 time-slots.

Now, in the scheduling scheme in which the disk access is fixed to the time-slot which is positioned a certain period of time away relatively from the reference time-slot, depending on a combination of the reference time-slots for already connected unit streams. there can be a case in which the reference time-slots for m pieces of new unit streams overlap with the reference time-slots for the already connected unit streams, no matter how these reference time-slots for m pieces of new unit streams are selected. In such a case, it is impossible to allocate the disk access to the time-slot, and it is impossible to connect a new stream.

In contrast, in the scheduling scheme in which the allocation position of the disk access is made variable among

Google Exhibit No. 1008
Page 253

the time-slots within the tolerable jitter range, as long as there is a vacant time-slot within the tolerable jitter range as in a case shown in Fig. 2, it suffices to allocate the disk access of a new unit stream to that vacant time-slot. Also, even when there is no vacant time-slot within the tolerable jitter range, by moving the disk access of the already connected unit stream to another time-slot within its tolerable jitter range, it is possible to create a vacant time-slot and allocate the disk access of a new unit stream to that vacated time-slot.

Even in this case, however, a new stream cannot be connected when it is impossible to create a vacant time-slot no matter how the disk accesses are moved. In order to lower a probability for such a situation to arise, it is preferable to select the reference time-slots such that the reference time-slots of the unit streams are not concentrated in a short period of time as much as possible, that is, the vacant time-slots are as widely spread as possible over all the time-slots.

As described, according to the present invention, the real time stream data with various data rates can be handled by using an appropriate number of independent unit streams according to a data rate of each real time stream data, and therefore it becomes possible to realize the disk access scheduling based on a single scheme regarding a time-slot interval, a block size, a block transfer period, and a buffer memory management, without distinguishing different data rates.

Also, according to the present invention, the stream resources are divided into amounts corresponding to the unit streams, and allocated as much as necessary according to a data rate of each real time stream data to be supplied, so that it becomes possible to utilize the stream resources efficiently without wasting any resource.

It is to be noted that, besides those already mentioned above, many modifications and variations of the above embodiments may be made without departing from the novel and advantageous features of the present invention. Accordingly, all such modifications and variations are intended to be included within the scope of the appended claims.

Claims

1. A real time stream server, comprising:

   entering means for entering real time stream data to be stored in the real time stream server;
   determining means for determining a number of unit streams to be used and a block transfer time for the real time stream data, according to a data rate of the real time stream data;
   dividing means for dividing the real time stream data into a plurality of blocks, each block being in a size to be transferred within the block transfer time, and sequentially distributing the blocks among as many unit streams as the number of unit streams to be used;
   a plurality of disk devices for sequentially storing the blocks of each unit stream data;
   a buffer memory for temporarily storing the blocks read out from said plurality of disk devices;
   control means for reading out the blocks constituting the real time stream data from said plurality of disk devices to the buffer memory, and reading out the real time stream data from the buffer memory, according to a request for the real time stream data from a client; and
   transfer means for transferring the real time stream data read out from the buffer memory to the client through a network.

2. The real time stream server of claim 1, wherein the control means includes:

   management means for managing a directory information for each real time stream data stored in said plurality of disk devices, the directory information indicating the number of unit streams to be used, a disk device ID number of a disk device which stores a top block of each unit stream, and a recording position of each block on a disk device which stores each block, for said each real time stream data.

3. The real time stream server of claim 1, wherein the control means carries out operations with respect to each real time stream data according to the directory information for each real time stream data managed in the management means.

4. The real time stream server of claim 1, wherein the control means includes:

   scheduling means for securing stream resources for as many unit streams as the number of unit streams to be used, scheduling a transfer start timing for each unit stream in order to transfer the blocks continuously in a sequential order to form the real time stream data, and controlling the transfer means to read out each block of each unit stream on the buffer memory within the block transfer time since the transfer start timing for each unit stream.

5. The real time stream server of claim 4, wherein the scheduling means schedules the transfer start timing for each

Google Exhibit No. 1008
Page 254

unit stream such that scheduled transfer start timings of the unit streams are displaced one another by the block transfer time.

6. The real time stream server of claim 1, wherein the unit stream is defined as a stream which is scheduled by using a prescribed block transfer period T, a prescribed block size L, and a prescribed time-slot interval I.

7. The real time stream server of claim 1, wherein the determining means detects a maximum data rate of the real time stream data, and determines the number m of unit streams to be used as a minimum integer which satisfies:

$$\text{the maximum data rate} \leqq m \times R$$

where R is a prescribed maximum data rate of each unit stream.

8. The real time stream server of claim 7, wherein the determining means determines the block transfer time as:

$$\text{the block transfer time} = T/m$$

where T is a prescribed block transfer period, and m is the number of unit streams to be used.

9. The real time stream server of claim 1, wherein the real time stream server has a capacity to supply Cmax sets of real time stream data with data rates not greater than R, where R is a prescribed maximum data rate of each unit stream, and the control means sets a number of supplies Cm (m = 1, 2, • • • • • • • •) to be allocated to each real time stream data with a data rate in a data rate range greater than (m-1)×R and not greater than m×R, arbitrarily within a range of:

$$\sum_{m}(m \times Cm) \leqq Cmax$$

where m is the number of unit streams to be used for said each real time stream data.

10. A method for operating a real time stream server having a plurality of disk devices and a buffer memory, comprising the steps of:

entering real time stream data into the real time stream server;
determining a number of unit streams to be used and a block transfer time for the real time stream data, according to a data rate of the real time stream data;
dividing the real time stream data into a plurality of blocks, each block being in a size to be transferred within the block transfer time, and sequentially distributing the blocks among as many unit streams as the number of unit streams to be used;
sequentially storing the blocks of each unit stream data into said plurality of disk devices;
controlling the real time stream server to read the blocks constituting the real time stream data from said plurality of disk devices to the buffer memory, and read out the real time stream data from the buffer memory, according to a request for the real time stream data from a client; and
transferring the real time stream data read out from the buffer memory to the client through a network.

11. The method of claim 10, wherein the controlling step includes the step of:

managing a directory information for each real time stream data stored in said plurality of disk devices, the directory information indicating the number of unit streams to be used, a disk device ID number of a disk device which stores a top block of each unit stream, and a recording position of each block on a disk device which stores each block, for said each real time stream data.

12. The method of claim 11, wherein the controlling step carries out operations with respect to each real time stream data according to the directory information for each real time stream data managed by the management step.

13. The method of claim 10, wherein the controlling step includes steps of:

Google Exhibit No. 1008
Page 255

securing stream resources for as many unit streams as the number of unit streams to be used;

scheduling a transfer start timing for each unit stream in order to transfer the blocks continuously in a sequential order to form the real time stream data; and

controlling the real time stream server to read out each block of each unit stream on the buffer memory within the block transfer time since the transfer start timing for each unit stream.

14. The method of claim 13, wherein the scheduling step schedules the transfer start timing for each unit stream such that scheduled transfer start timings of the unit streams are displaced one another by the block transfer time.

15. The method of claim 10, wherein the unit stream is defined as a stream which is scheduled by using a prescribed block transfer period T, a prescribed block size L, and a prescribed time-slot interval l.

16. The method of claim 10, wherein the determining step detects a maximum data rate of the real time stream data, and determines the number m of unit streams to be used as a minimum integer which satisfies:

$$\text{the maximum data rate} \leqq m \times R$$

where R is a prescribed maximum data rate of each unit stream.

17. The method of claim 16, wherein the determining step determines the block transfer time as:

$$\text{the block transfer time} = T/m$$

where T is a prescribed block transfer period, and m is the number of unit streams to be used.

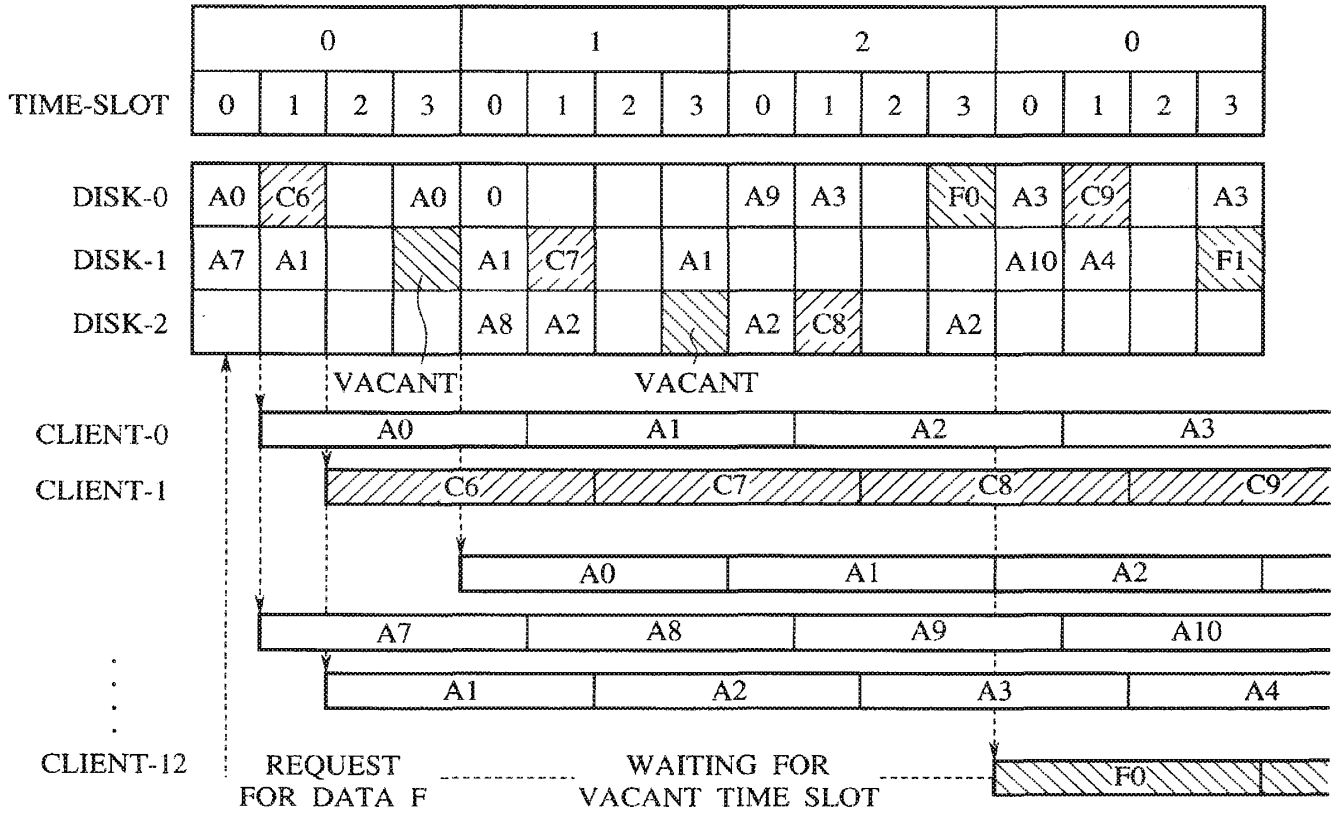
18. The method of claim 10, wherein the real time stream server has a capacity to supply Cmax sets of real time stream data with data rates not greater than R, where R is a prescribed maximum data rate of each unit stream, and the controlling step sets a number of supplies Cm (m = 1, 2, ・・・・・・・・・) to be allocated to each real time stream data with a data rate in a data rate range greater than (m-1)×R and not greater than m×R, arbitrarily within a range of:

$$\sum_{m}(m \times Cm) \leqq Cmax$$

where m is the number of unit streams to be used for said each real time stream data.

Google Exhibit No. 1008
Page 256

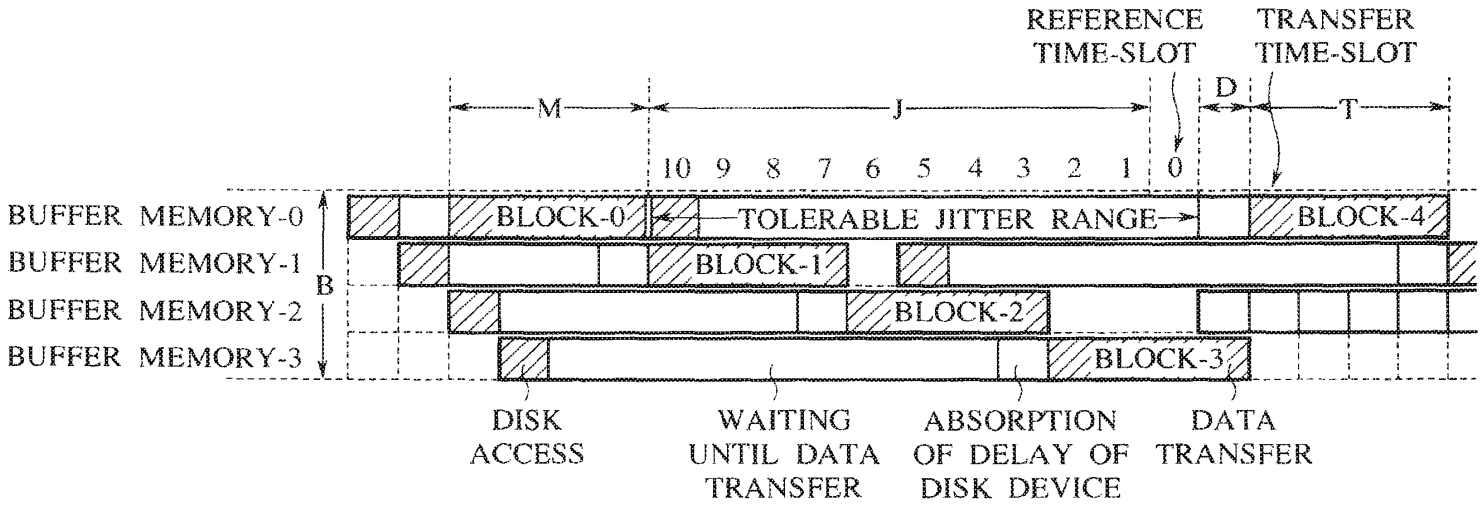
FIG.1
PRIOR ART



EP 0 762 300 A2

Google Exhibit No. 1008
Page 257

FIG.2
PRIOR ART



EP 0 762 300 A2

Google Exhibit No. 1008
Page 258

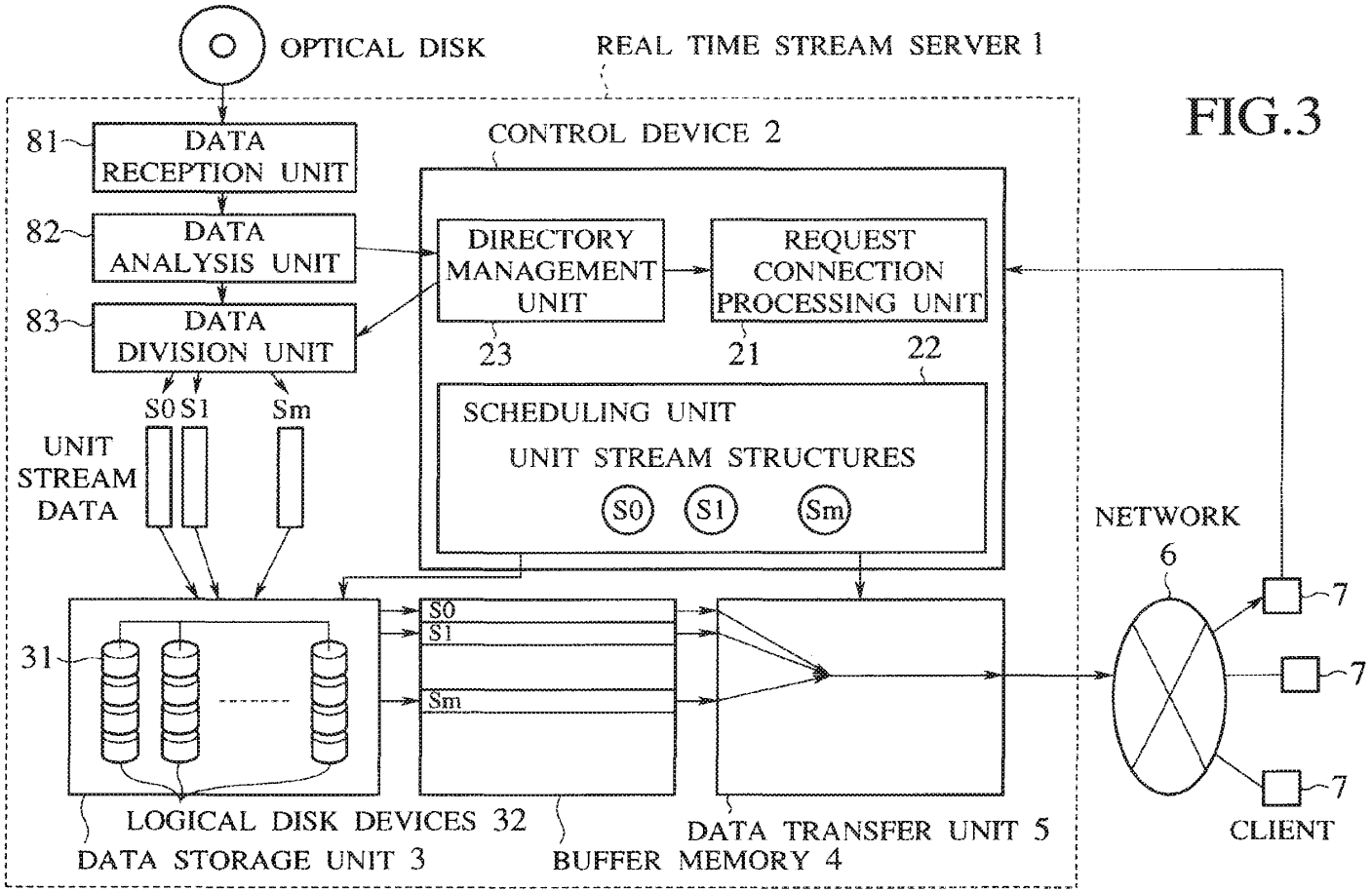
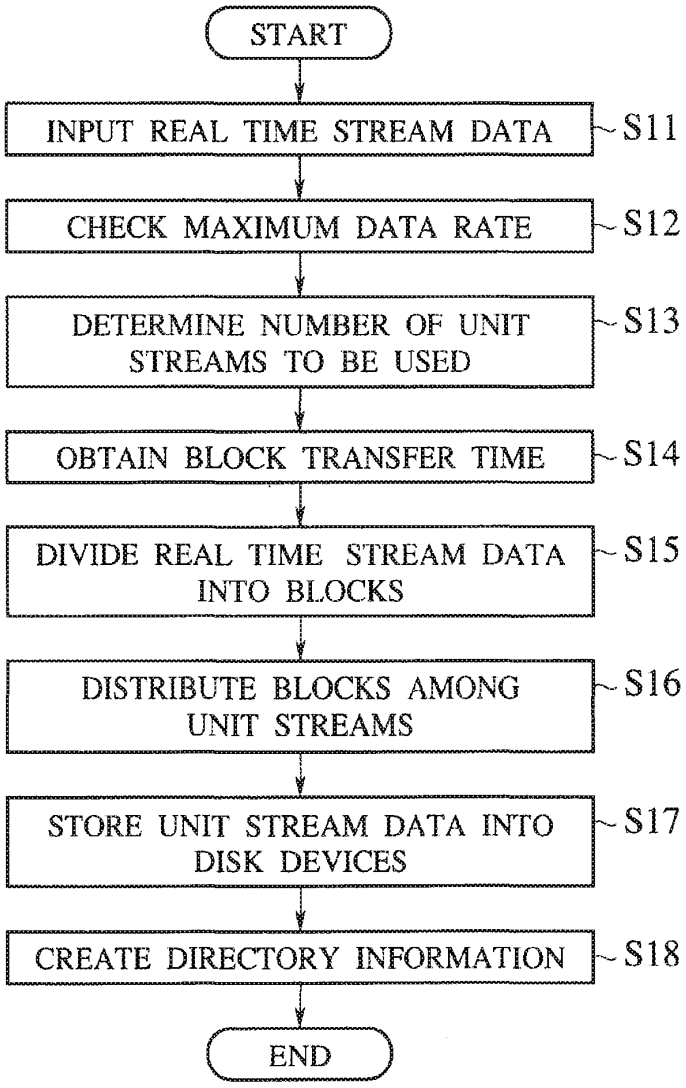


FIG.3

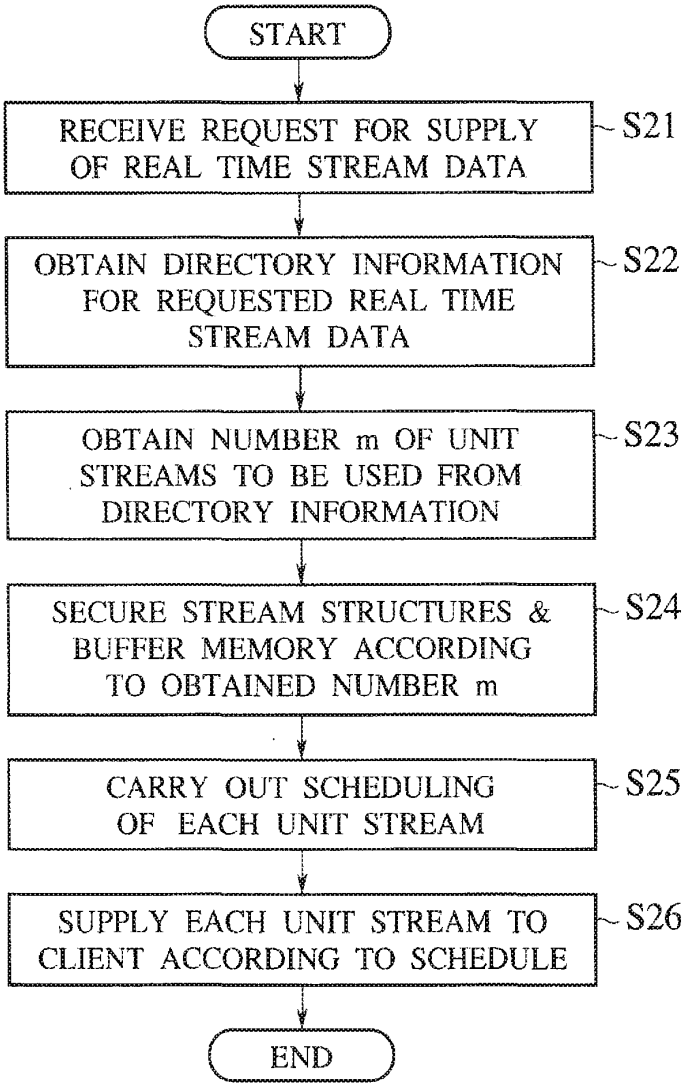
EP 0 762 300 A2

FIG.4



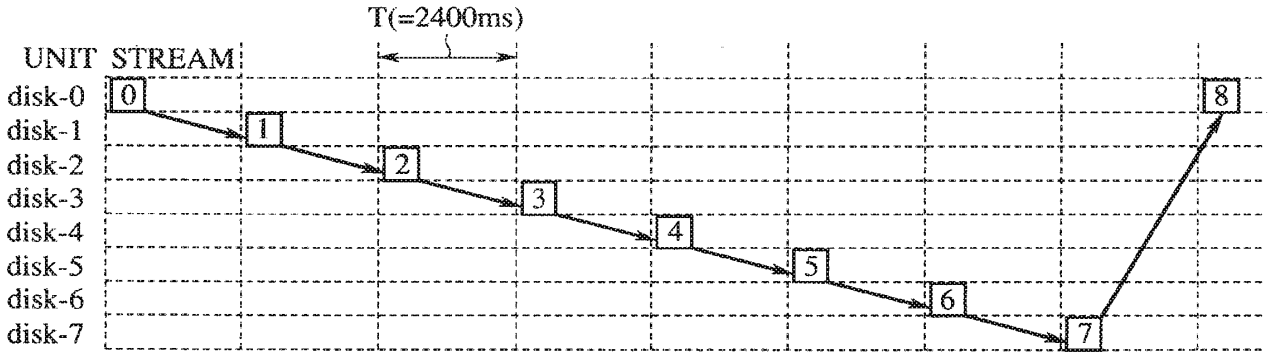
Google Exhibit No. 1008
Page 260

FIG.5



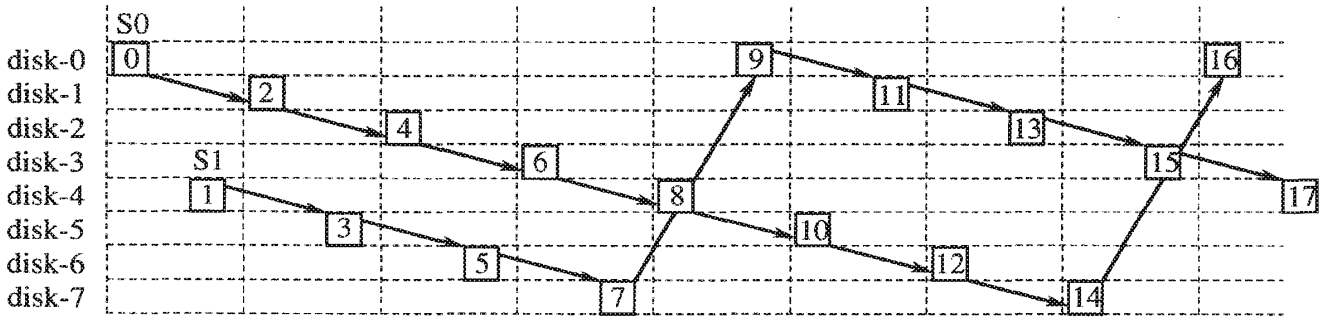
Google Exhibit No. 1008
Page 261

FIG.6



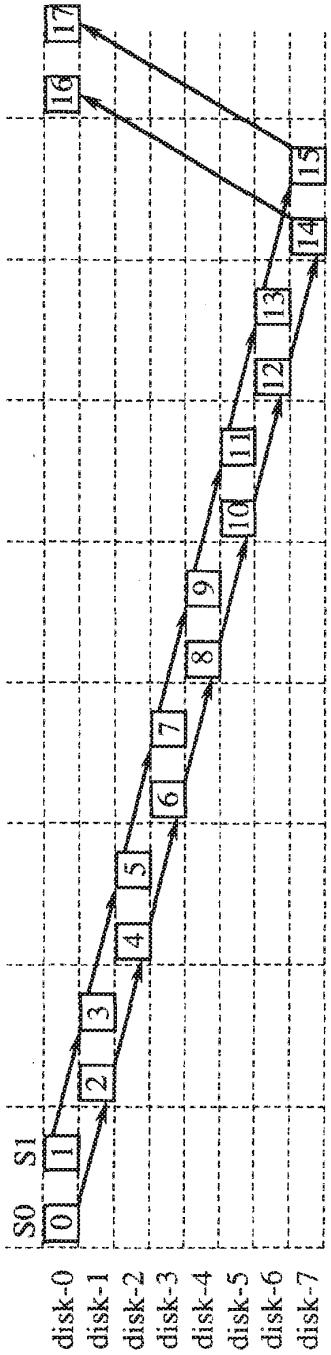
Google Exhibit No. 1008
Page 262

FIG.7



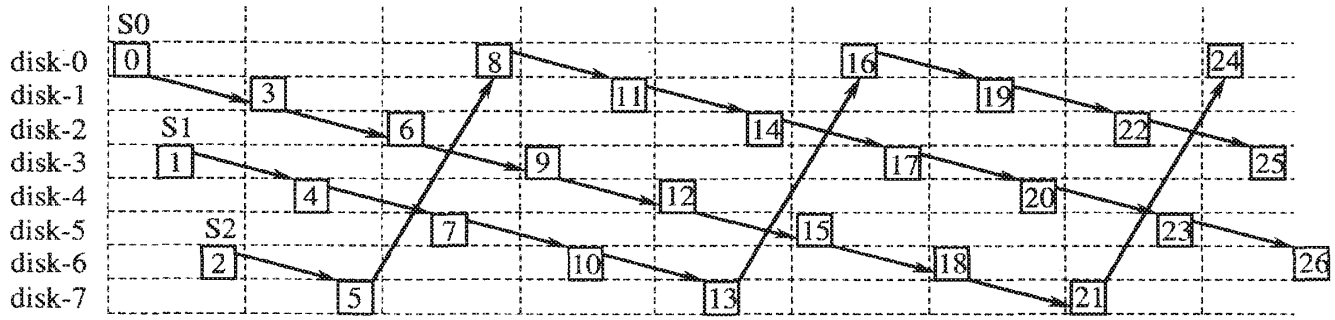
Google Exhibit No. 1008
Page 263

FIG.8



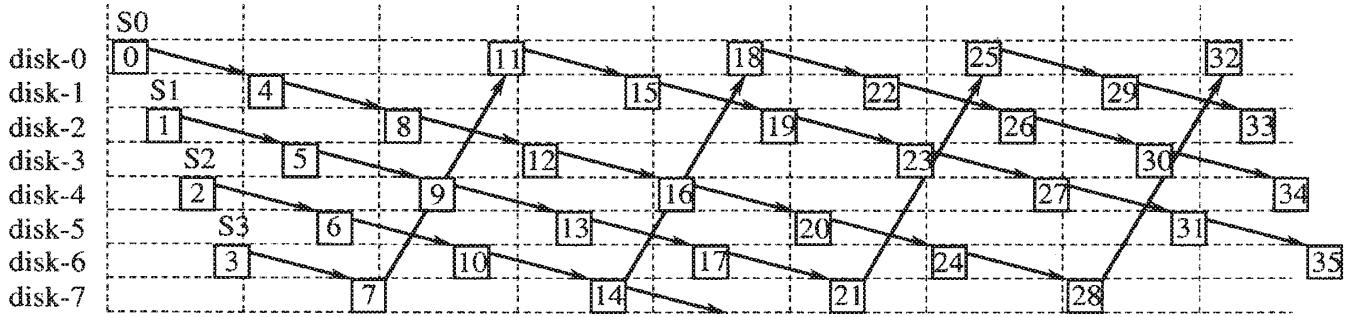
Google Exhibit No. 1008
Page 264

FIG.9



Google Exhibit No. 1008
Page 265

FIG.10


S0
disk-0
0
11
18
25
32
disk-1
S1
4
15
22
29
disk-2
1
8
19
26
33
disk-3
S2
5
12
23
30
disk-4
2
9
16
27
34
disk-5
S3
6
13
20
31
disk-6
3
10
17
24
35
disk-7
7
14
21
28

Electronic Acknowledgement Receipt

EFS ID:	37154505
Application Number:	16181285
International Application Number:	
Confirmation Number:	1045
Title of Invention:	Multimedia System For Mobile Client Platforms
First Named Inventor/Applicant Name:	Mark Sinclair KREBS
Customer Number:	132285
Filer:	Yasin Bismilla/Eddie Rowell
Filer Authorized By:	Yasin Bismilla
Attorney Docket Number:	2018-00816
Receipt Date:	13-SEP-2019
Filing Date:	05-NOV-2018
Time Stamp:	15:22:40
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

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)	2018-00816_IDS.pdf	613886 d259d0982f30d307cf8d555bbde5fa33198da837	no	8

Warnings:

Information:					
2	Foreign Reference	WO9730551A1.pdf	13283248 9c018da2664a45e64cedfcfb72aa707cb5f3ffc1	no	98
Warnings:					
Information:					
3	Foreign Reference	EP0762300A2.pdf	2641051 8e6d1c9123640ed782953e826bd41d925dcc441c	no	21
Warnings:					
Information:					
4	Non Patent Literature	IIS_Smooth_Streaming_Technical_Overview.pdf	1188005 41d7cd0b3da148bc23425ea15d25551053dc7d32	no	17
Warnings:					
Information:					
5	Non Patent Literature	Two_Decades_of_Internet_Video_Streaming.pdf	1308863 207a5921d13e62c6655d9ec1ac4fc28bfa3af216	no	20
Warnings:					
Information:					
6	Non Patent Literature	A_brief_history_of_the_object-oriented_approach.pdf	408798 9b20c11fa7ae1c0c43818c48ed2c73360f77d8d7	no	11
Warnings:					
Information:					
7	Non Patent Literature	Article_on_Easter_Egg_Roll.pdf	163039 c89d6d7569c318eb6f6d4d8c954dd20ae9d5b0ac	no	3
Warnings:					
Information:					
8	Non Patent Literature	RealPlayerG2Streaming.pdf	957224 4ba23c1b2880b4d5d4380fbe7b7ff8e79fc4f25c	no	130
Warnings:					
Information:					
		Total Files Size (in bytes):	20564114		

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.

Doc code: IDS
Doc description: Information Disclosure Statement (IDS) Filed

PTO/SB/08a (01-10)
Approved for use through 07/31/2012. OMB 0651-0031
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.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (**Not for submission under 37 CFR 1.99)**		
	Application Number	16181285
	Filing Date	2018-11-05
	First Named Inventor	Mark Sinclair KREBS
	Art Unit	2488
	Examiner Name	DOBBS, KRISTIN SENSMEIER
	Attorney Docket Number	2018-00816

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	9800538	B2	2017-10-24	Miraj Mostafa	

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	20010042107	A1	2001-11-15	Stephen R. Palm	
	2	20030204605	A1	2003-10-30	Hudson et al.	
	3	20050076136	A1	2005-04-07	Cho et al.	

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] i	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1	0193161	WO	A1	2001-12-06	Zebus Group, Inc.		

EFS Web 2.1.17

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	16181285
	Filing Date	2018-11-05
	First Named Inventor	Mark Sinclair KREBS
	Art Unit	2488
	Examiner Name	DOBBS, KRISTIN SENSMEIER
	Attorney Docket Number	2018-00816

	2	1411698	EP	A1	2004-04-21	Apple Computer, Inc.		
	3	02082292	WO	A1	2002-10-17	Playstream, L.L.C.		

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T5
	1	LI ET AL., "Distributed Multimedia Systems," Proceedings of the IEEE, July 1997, pgs. 1063-1108,, Vol. 85 No. 7	
	2	FRANC KOZAMERNIK, "Media Streaming over the internet", EBU Technical Review, October 2002, pgs 1-15, Webcasting	
	3	ZHANG ET AL., "Resource Allocation for Multimedia Streaming Over the Internet," IEEE Transactions on Multimedia, September 2001, Vol. 3. No. 3, pgs. 339-355	
	4	KRASIC ET AL., "The Case for Streaming Multimedia with TCP," Lecture Notes in Computer Science, August 2001, pgs. 1-6, Vol. 2158	
	5	SHAHABI ET AL., "Yima: A Second-Generation Continuous Media Server" Computer, June 2002, pgs. 56-62, Vol 35 No. 6, IEEE	

If you wish to add additional non-patent literature document citation information please click the Add button Add

EXAMINER SIGNATURE

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	16181285
	Filing Date	2018-11-05
	First Named Inventor	Mark Sinclair KREBS
	Art Unit	2488
	Examiner Name	DOBBS, KRISTIN SENSMEIER
	Attorney Docket Number	2018-00816

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	16181285
	Filing Date	2018-11-05
	First Named Inventor	Mark Sinclair KREBS
	Art Unit	2488
	Examiner Name	DOBBS, KRISTIN SENSMEIER
	Attorney Docket Number	2018-00816

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

See attached certification statement.

The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☒ A certification statement is not submitted herewith.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Yasin Bismilla/	Date (YYYY-MM-DD)	2019-11-22
Name/Print	Yasin Bismilla	Registration Number	64,756

This collection of information is required by 37 CFR 1.97 and 1.98. 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 1 hour 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.**

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 record s.

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 Cooperation Treaty.

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.

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 December 2001 (06.12.2001)

PCT

(10) International Publication Number
WO 01/93161 A1

(51) International Patent Classification⁷: G06F 17/60

(21) International Application Number: PCT/US01/17259

(22) International Filing Date: 29 May 2001 (29.05.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/579,368 26 May 2000 (26.05.2000) US
60/274,138 8 March 2001 (08.03.2001) US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier application:
US 09/579,368 (CIP)
Filed on 26 May 2000 (26.05.2000)

(71) Applicant *(for all designated States except US)*: **ZEBUS GROUP, INC.** [US/US]; Suite 320, 1 Woodbridge Center Drive, Woodbridge, NJ 07095 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **BHUTRA, Chand** [US/US]; 63 Forest Glen Drive, Highland Park, NJ 08904 (US). **VARMA, Praveen** [IN/US]; 909 Kings Court, Woodbridge, NJ 07095 (US).

(74) Agent: **HOFFBERG, Steven, M.**; Milde, Hoffberg & Macklin, LLP, Suite 460, 10 Bank Street, White Plains, NY 10606 (US).

(81) Designated States *(national)*: 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.

(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.

WO 01/93161 A1

(54) Title: ONLINE MULTIMEDIA SYSTEM AND METHOD

(57) Abstract: A system and method for delivering a targeted advertisement, comprising means for performing the steps of classifying a consumer by assigning to the consumer a behavior classification based on a set of expressly defined set of preferences, an antomatically derived set of preferences, and a prior purchasing behavior; selecting a targeted advertisement to be delivered to the consumer, based on the classification; presenting the targeted advertisement to the consumer; and monitoring an action immediately subsequent to the presentation of the targeted advertisement. According to another embodiment, a system and method are provided for delivering a streaming multimedia advertisement to a consumer, separately from streaming multimedia content, and presenting these seamlessly and without intervening delay.

Google Exhibit No. 1008
Page 275

ONLINE MULTIMEDIA SYSTEM AND METHOD

FIELD OF THE INVENTION

The present invention relates to the field of online multimedia systems and methods, and more particularly to improved presentation methods, systems and architectures.

BACKGROUND OF THE INVENTION

Consumer media and content services are competitive businesses. Thus, they seek to deliver the highest quality content and/or service (i.e., consumer benefit) for the lowest perceived consumer price. In part due to the perceived vale, consumers typically spend a large portion of their time experiencing and interacting with such media. Thus, as is well known, commercial subsidies for media content and delivery often maintain low cost while promoting advertiser's brands cost effectively. Further, such advertising, if efficiently targeted, may itself comprise value to the consumer.

A growing trend has been to try to integrate consumer commercial messages with media content, in order to closely link the commercial sponsorship with the media, and to prevent segregation of the commercial message from the media content. However, this integration limits the ability to personally target messages and separate sponsorship for various presentations of the media.

Typically, subsidies for content and services made publicly available on the Internet has been through commercial advertisements, in the form of static or crudely animated graphics. Compensation by the advertisers typically are dependent on impressions (presentation to a user), click-throughs (minimal interaction with a user), or network marketing (contingent payment based on consummation of a transaction with the user). Thus, these commercial models are most appropriate for electronic commerce vendors (e.g., retailers), and less desired for standard consumer brands sold through "bricks and mortar" distribution channels.

Multimedia advertising is considered more effective, and therefore more valuable, in establishing and promoting consumer brands, as compared to static graphic ads, such as banner ads and print ads. These multimedia ads both potentially convey a large amount of information in an entertaining and relatively passive format, and allow the producer of the ad to define a mood and tempo. Thus, multimedia advertisements of 15-120 seconds are highly valued for marketing of goods and promotion of brands.

A number of multimedia application developers have proposed delivery of advertisements in multimedia form, such as television-type commercials. Generally, these applications, which include Microsoft Windows Media Player, Real Networks Realplayer and Apple Corporation Quicktime, are suitable for the presentation of advertisements, but fail to

address the user interaction and motivational aspects for accepting and viewing the advertisement. See, "Understanding the Advertising Application for the Realsystem G2", Real Networks, Inc.; Wang, Nelson, "Software Allows Streaming Ads Without Plug-in", InternetWorld, August 25, 1997; www.narrative.com/technology.htm (and referenced pages, re: Enliven rich media advertising solution), expressly incorporated herein by reference. Simply, without appropriate incentive, even users with broadband Internet connectivity, and especially those without, are unlikely to knowingly request a multimedia advertisement file from a remote server, at the cost of seconds or minutes of download latency, followed by a period for presentation, with accompanying usage of connection bandwidth. Likewise, without appropriate incentivization, users are also unlikely to knowingly accept a "push" of such advertising to them. Thus, the coordination of media content and commercials is preferably coordinated according to presenter-defined rules, rather than promoted by the receiver. Thus, while the concept of pulling or pushing multimedia advertisements is known, the issue of how to make such advertisements acceptable to the user remains, in the prior art, unresolved.

According to present technologies for switched packet networks such as the Internet, advertising is either directly and statically integrated with the media, or presented as separate transmissions, each of which requires a pre-buffering latency.

A particular advantage of the Internet media distribution is that updates may be made nearly instantaneously, and information may be substituted or customized based on an identification of the user, chronology, or otherwise. Thus, a user may be identified, and content specifically organized and presented to the user, based on both the user's request and the user's identification. A standard scheme for automatic identification and/or tracking of users exists, called an Internet browser cookie, or simply cookie, which provides a relatively small file on the user's computer created by the remote server and retrievable by the remote server. Cookies may be persistent, or expire at predetermined times, allowing permanent identification or simple session tracking.

The following patents, expressly incorporated herein by reference in their entirety, disclose aspects of network media delivery: United States Patent Nos. 6,199,082 (Ferrel, et al., March 6, 2001); 6,199,076 (Logan, et al., March 6, 2001), 6,198,920 (Doviak, et al., March 6, 2001), 6,185,625 (Tso, et al., February 6, 2001), 6,177,930 (Chernock, et al., January 23, 2001), 6,173,271 (Goodman, et al., January 9, 2001), 6,154,771 (Rangan, et al., November 28, 2000), 6,144,944 (Kurtzman, II, et al., November 7, 2000), 6,161,137 (Ogdon, et al., December 12, 2000), 6,094,677 (Capek, et al., July 25, 2000), 6,119,163 (Monteiro, et al., September 12,

2000), 6,108,703 (Leighton, et al., August 22, 2000), 6,101,180 (Donahue, et al., August 8, 2000), and 6,014,634 (Scroggie, et al.).

Advertisers often desire to target particular audiences for their advertisements. These targeted audiences are the audiences that an advertiser believes is most likely to be influenced by the advertisement or otherwise provide revenues or profits. By selectively targeting particular audiences the advertiser is able to expend his or her advertising resources in an efficient manner. Thus, it may be efficient to provide a system that allows electronic advertisers to target specific audiences, and thus not require advertisers to provide an single advertisement to the entire population. Currently, advertisers are able to implement a limited form of targeted advertising over the Internet. For example, Doubleclick implements such a method. This is accomplished by sending a block of data, such as a "cookie," from a remote host or server (i.e., a Web server) maintained by an advertiser to a computer (i.e., a client system) that has access to the remote server via the World Wide Web. The cookie can be used to identify the client system (by identifying the Web browser) and to instruct the server to send a customized copy of the requested Web page to the Web browser. Since cookies are also used to track a consumer's online activity, a Web server can deliver targeted advertisements to a consumer's Web browser, based on the consumer's online activity, as well as to compile information regarding the consumer's tastes and tendencies.

Advertisers are generally willing to pay more to deliver an impression (e.g., a banner ad or other type of advertisement) to users who are especially sensitive to advertisements for their products or are seeking to purchase products corresponding to those sold by the advertisers or are within a target demographic category, and the economic model often provides greater compensation in the event of a "click through", which is a positive action taken by the user to interact with the ad to receive further information. This principle, of course, actually operates correspondingly in traditional media. For example, a gardening tool manufacturer in generally is willing to pay more per subscriber to place advertisements in a magazine having content directed to gardeners than in a general interest or urban-directed publication. On the other hand, in an incentive advertising scheme, the publisher receives compensation or commission on a concluded transaction, and therefore the publisher has an incentive to properly target the advertisements in order to maximize compensation.

Accordingly, a conditional probability of a subsequent action by the user may be assessed for each interaction, and that, on the basis of that probability, an economic parameter may be altered. Thus, for example, the selection of a hyperlink by the user through a browser may be associated with a calculated probability that the user will subsequently purchase a good

or service. This probability may then be used to calculate an advertiser charge for delivery of an advertisement, or to prioritize the advertisements sent to the user in order to, for example, maximize the utility to the selected advertiser, the advertisement serving system operator, to the user, or some combination thereof. This calculated probability may also be used to adapt the information subsequently presented to the user. This probability may be calculated, for example, based on a population statistic plus a recent history of the particular user, a collaborative filtering scheme, a long-term monitoring of the user through the use, for example, of cookies and a database, or other scheme, or through express input of user characteristics, such as demographic profile, survey response, or a direct user communication. The logic used to predict the probability may be formal Bayesean, fuzzy logic, a multiple regression equation, neural networks, or other known logic.

The following references, each expressly incorporated herein by reference in its entirety, relate to user profiling or personalized media delivery: U.S. Patent Nos. 5,724,521 (Dedrick), 5,774,357 (Hoffberg et al.), 5,890,152 (Rapaport et al.), 4,602,279 (Freeman), 5,283,731 (Lalonde, et al.), 5,794,210 and 5,855,008 (Goldhaber, et al.), 5,974,398 (Hanson), 5,155,591 (Wachob), 5,191,410 (McCalley, et al.), 5,305,195 (Murphy), 5,201,010 (Deaton, et al.), 5,515,098 (Carles), 5,948,061 (Merriman, et al.), 6,006,197 (d'Eon, et al.), 6,009,409 (Adler, et al.), 5,893,075 (Plainfield, et al.), 5,937,392 (Alberts), 5,887,243 (Harvey et al.), 5,872,588 (Aras et al.), 5,774,170 (Hite et al.), 5,636, 346 (Saxe), 5,870,724 (Lawlor et al.), 5,724,424 (Gifford), 5,838,314 (Neel, et al.), 5,231,494 (Wachob), 5,974,398 (Hanson, et al.), 5,933,811 (Angles, et al.), 6,006,265 (Rangan, et al.), 6,005,561 (Hawkins, et al.), 6,009,410 (LeMole et al.), 6,053,554 (Hendricks, et al.), 5,991,740 (Messer), and 5,227,874 (Von Kohorn).

In recent years, the field of data mining, or extracting useful information from bodies of accumulated raw data, has provided a fertile new frontier for database and software technologies. Methods for mining transaction databases to discover association rules have been disclosed in Agrawal et al., "Mining Association Rules Between Sets of Items in Large Databases", Proc. of the ACM SigMod Conf. on Management of Data, May 1993, pp. 207-216, and in Houtsma et al., "Set-Oriented Mining of Association Rules", IBM Research Report RJ 9567, October, 1993. See also, U.S. Pat. Nos. 5,615,341, 5,794,209, 5,742,811, 5,724,573, 5,819,266, 5,842,200, 6,061,682; 5,812,997 5,796,209, 5,724,573 5,812,997 (Agrawal et al.), and 5,884,305 (Kleinberg, et al.), expressly incorporated herein by reference.

User modeling means to create a model of the user that contains information about the user that is relevant for a particular system. Thus, the user modeling system seeks to define sufficient characteristics of the user to determine the prospective actions or preferences of the

user, and employ these characteristics to make predictions. Often, the user modeling system is used interactively with the user, facilitating the use of the system by intelligently predicting the user's inputs. The scope of the user model may include, for example, characteristics of the user which are independent of content, such as language, reading level, fields of expertise, physical impairments, and the like, as well as content specific characteristics, such as the user's taste and interests for motion picture entertainment, for example as part of a film recommending system, or the user's knowledge of a given academic subject, for an educational or testing system. User preferences may be time dependent, and therefore diurnal or seasonal variations may be important factors in defining an accurate model of the user, i.e., the predicting the intent and/or desires of the user in a respective context. Different systems use different techniques for constructing and implementing a user model. The simplest and most straightforward is a technique of a user survey, requiring some dedicated activity of the user toward defining the user model. A second technique monitors the activities of the user to detect patterns and actions indicative of user characteristics. Another way of creating a user model is through the use of collaborative filtering, in which the user typically identifies himself or herself with a class of users, by predefined or adaptive categories, to classify the user into a group with common characteristics of interest. Feedback may also be employed to deduce user preferences and refine adaptive models. See, Greer, J. E., & McCalla, G. I. (Eds.): "Student Modeling: The Key to Individualized Knowledge-Based Instruction" NATO ASI Series F Vol. 125 (1993) Berlin: Springer-Verlag; Gaines, Brian R., and Shaw, Mildred L.G., "Concept Maps as Hypermedia Components", (Internet); Akoulchina, Irina, and Ganascia, Jean-Gabriel, "SATELIT-Agent: An Adaptive Interface Based on Learning Agents Interface Technology", In Anthony Jameson, Cecile Paris and Carlo Tasso (Eds), User Modeling: Proc. Of the Sixth Intl. Conf. UM97, Vienna, New York: Springer Wein, New York (1997); Benaki, Eftihia, Karkaletis, Vangelis A., Spyropoulos, Constantine D, "Integrating User Modeling Into Information Extraction: The UMIE Prototype", In Anthony Jameson, Cecile Paris and Carlo Tasso (Eds), User Modeling: Proc. Of the Sixth Intl. Conf. UM97, Vienna, New York: Springer Wein, New York (1997); Maglio, Paul P., and Barret, Rob, "How To Build Modeling Agents to Support Web Searchers" In Anthony Jameson, Cecile Paris and Carlo Tasso (Eds), User Modeling: Proc. Of the Sixth Intl. Conf. UM97, Vienna, New York: Springer Wein, New York (1997); Hohl, H., Böcker, H., Gunzenhäuser R.: "Hypadapter: An adaptive hypertext system for exploratory learning and programming", User Modeling and user adapted Interaction 6, 2-3, (1996) 131-156; Höök, K., Karlgren, J., Waern, A., Dahlbäck, N., Jansson, C.G., Karlgren, K. and Lemaire, B.: "A glassbox approach to adaptive hypermedia"; User Modeling and User-Adapted Interaction, 6, 2-3, (1996)

157-184; Boyle C. and Encarnacion A.O.: "MetaDoc: an adaptive hypertext reading system"; User modeling and User-Adapted Interaction, 4 (1994) 1-21; Brusilovsky, P., Eklund, J.: "A study of user model based link annotation in educational hypermedia"; Journal of Universal Computer Science, Vol. 4 No 4 (1998) 429-448; Chin, D.: "User Modeling in UC: the Unix Consultant"; Proceedings of the CHI-86 Conference, Boston (1986); Moore, J.D. & Swartout, W.R.: "Pointing: A way toward explanation dialogue"; Eight National Conference on Artificial Intelligence, (1989) 457-464; [AVANTI homepage] http://zeus.gmd.de/projects/avanti.html; http://zeus.gmd.de/UM97/Fink/Fink.html; Fink, J., Kobsa, A., Schreck, J.: "Personalized hypermedia information provision through adaptive and adaptable system features: User modeling, privacy and security issues"; Eftihia Benaki, Vangelis A. Karkaletsis, Constantine D. Spyropoulos, "Adaptive Systems and User Modeling on the World Wide Web", Proceedings of the workshop, Sixth International Conference on User Modeling, Chia Laguna, Sardinia, 2-5 June 1997; Brajnik, G., Guida, G., Tasso, C., (1990): User modeling in Expert Man-Machine Interfaces: A case study in Intelligent Information Retrieval, in IEEE Transactions on systems, man, and cybernetics, 20:166-185; Brajnik Giorgio and Carlo Tasso, (1994): A shell for developing non-monotonic user modeling systems in International Journal of Human Computer Studies, 40:31-62; Croft, B. and Thompson, R., (1986): An overview of the IR Document Retrieval System, in Proceedings of the 2nd Conference on Computer Interfaces and Interaction for Information Retrieval; Karkaletsis, E., Benaki, E., Spyropoulos, C., Collier, R., (1996): D-1.3.1: Defining User Profiles and Domain Knowledge Format, ECRAN; Kay, J., (1995): The um toolkit for Cooperative User Modeling, in User Modeling and User-Adapted Interaction, 4:146-196; Jon Orwant, (1993): Doppelganger Goes to School: Machine Learning for User Modeling, M.Sc. thesis at MIT; J. Orwant, "For want of a bit the user was lost: Cheap user modeling", MIT Media Lab, Vol. 35, No. 3&4 (1996); Rich, E., (1983): "Users are individuals: individualising user models", International Journal of Man-Machine Studies, 18:199-214. See, 5,966,533 (Moody), 6,055,573 (Gardenswartz, et al.), 5,819,285 (Damico, et al.), 6,012,051 (Sammon, et al.), 6,006,218 (Breese, et al.), 6,012,052 (Altschuler, et al.), 6,014,638 (Burge, et al.), 5,991,735 (Gerace, et al.), 5,978,766 (Luciw), 5,977,964 (Williams, et al.), 5,974,412 (Hazelhurst, et al.), 5,963,645 (Kigawa, et al.), 5,801,747 (Bedard), 5,758,259 (Lawler), 5,945,988 (Williams, et al.), 6,005,597 (Barrett, et al.), 5,973,683 (Cragun, et al.), 5,946,490 (Lieberherr, et al.), 5,446,919 (Wilkins), 4,970,681 (Bennett), 5,710,887 (Chelliah, et al.), 5,438,355 (Palmer), 5,895,454 (Harrington), 5,920,477 (Hoffberg et al.), PCT/US99/65237 (Metabyte Inc., www.mbtv.com), each of which is expressly incorporated herein by reference.

Micropayments are often preferred where the amount of the transaction does not justify the costs of complete financial security. In the micropayment scheme, typically a direct communication between creditor and debtor is not required; rather, the transaction produces a result which eventually results in an economic transfer, but which may remain outstanding subsequent to transfer of the underlying goods or services. The theory underlying this micropayment scheme is that the monetary units are small enough such that risks of failure in transaction closure is relatively insignificant for both parties, but that a user gets few chances to default before credit is withdrawn. On the other hand, the transaction costs of a non-real time transactions of small monetary units are substantially less than those of secure, unlimited or potentially high value, real time verified transactions, allowing and facilitating such types of commerce. Thus, the rights management system may employ applets local to the client system, which communicate with other applets and/or the server and/or a vendor/rights-holder to validate a transaction, at low transactional costs.

The following U.S. Patents, expressly incorporated herein by reference, define aspects of micropayment, digital certificate, and on-line payment systems: 5,930,777 (Barber); 5,857,023 (Demers et al.); 5,815,657 (Williams); 5,793,868 (Micali); 5,717,757 (Micali); 5,666,416 (Micali); 5,677,955 (Doggett et al.); 5,839,119(Krsul; et al.); 5,915,093 (Berlin et al.); 5,937,394 (Wong, et al.); 5,933,498 (Schneck et al.); 5,903,880 (Biffar); 5,903,651 (Kocher); 5,884,277 (Khosla); 5,960,083 (Micali); 5,963,924 (Williams et al.); 5,996,076 (Rowney et al.); 6,016,484 (Williams et al.); 6,018,724 (Arent); 6,021,202 (Anderson et al.); 6,035,402 (Vaeth et al.); 6,049,786 (Smorodinsky); 6,049,787 (Takahashi, et al.); 6,058,381 (Nelson); 6,061,448 (Smith, et al.); 5,987,132 (Rowney); and 6,061,665 (Bahreman). See also, Rivest and Shamir, "PayWord and MicroMint: Two Simple Micropayment Schemes" (May 7, 1996), expressly incorporated herein by reference; Micro PAYMENT transfer Protocol (MPTP) Version 0.1 (22-Nov-95) et seq, http://www.w3.org/pub/WWW/TR/WD-mptp; Common Markup for web Micropayment Systems, http://www.w3.org/TR/WD-Micropayment-Markup (09-Jun-99); "Distributing Intellectual Property: a Model of Microtransaction Based Upon Metadata and Digital Signatures", Olivia, Maurizio, http://olivia.modlang.denison.edu/~olivia/RFC/09/.

It is also noted that, while a user may have to account for rights of third parties in order to use content, advertiser or sponsor subsidies may inure to the user based on viewing or access of advertisements, click-throughs, or the like. Therefore, while funds transfers may be necessary vis-à-vis sponsors, service providers, and rights-holders, the process may be transparent and potentially without net cost to the user. In this case, the risks of tampering and fraud based on manipulation of client-system resident data and applications will generally be low, and therefore

the level of security and protection of these client-resident data and applets may also be low. However, as deemed necessary, a high degree of security may be implemented, such as employing a hardware "dongle", secure encryption and watermarking techniques, or real-time transaction verification. There are a number of pay-per-view (PPV) systems known; however, as a necessary element of security, a proprietary or encrypted media stream is required to enforce rights limitations. See, U.S. Patent Nos. 6,057,872 (Candelore), expressly incorporated herein by reference.

The prior art streaming media advertising technology can deliver (a) banner advertisements located around the periphery of the video stream; (b) Banner ads seen before a stream starts, (c) Streaming video commercials using a play list approach that inserts commercials at the beginning or at the end of the stream (This approach, while becoming common, is severely limited due to its lack of targeting capabilities and significant buffering delays.) and (d) non-targeted commercials inserted in a live broadcast, again using a play list approach. The prior art technologies or business processes do not exist that can insert a targeted commercial in a web page, in a pop-up window, in a video stream (live or broadcast) at any insertion point from a single ad serving system. The current ad serving systems are disjoint, meaning that banner ads are served from one source (i.e. Doubleclick), interstitials are served from another source (i.e. Unicast), video ads are served from another source (e.g., ibeam, or Akamai) and audio ads are served from another separate source (as served by Hiwire and Lighteningcast). This approach is inherently limited by design. Specifically, a system designed to serve a certain type of media cannot be reconfigured without a significant cost and time. The companies that have created video ad serving technologies require clients to host all of their video content (advertisements and existing video clips) at the company's hosting facilities. Also, in order for these companies to insert advertisements into existing video streams, they must edit the video clip to include the advertisement in one file. Since everyone ends up watching the same commercial, the return on investment on ad dollars is severely limited.

SUMMARY OF THE INVENTION

The current convergence of multimedia formats necessitates an ad serving system that can serve ads in any format depending on the particular call/request made by a web site or a user. The preferred embodiment of the system according to the present invention provides a system that serves ads in every media format, regardless of connection speed, browser type and player type. It therefore allows advertisers to provide each user with the most sophisticated and effective media type (interactive audio/video commercials) based on the user's machine and network capabilities. The system according to the present invention also allows advertisers to efficiently manage their advertising campaigns.

The present invention offers a platform that inserts a multimedia commercial (audio/video) with full interactivity that can be completely server side and database driven, meaning that it has the ability to serve any type of media to the web page at any time. This commercial is either streamed directly from the server (Broadband users) or accessed from the hard drive where ads have been pre-cached in a particular order using a software process (narrowband users). The server side processes know where the advertisement must be shown. The commercial is delivered with the desired interactivity specified by the advertiser. This interactivity is completely customizable and dynamically modifiable, as it is a server side process and therefore under central control. Thus, the present invention is not limited to broadcast media-style commercials, and the entire presentation may be interactive and/or adaptive. Alternately, the interactivity may be provided externally to a linear multimedia presentation, through, for example, the control application, a web browser or graphic user interface elements.

According to a preferred embodiment, before a audio/video ad commences, the advertising location/space on the web page is filled with a base media (gif or animated gif) to avoid a black space. Upon the completion of the commercial, the ad space converts back to the gif/animated gif, thus offering multiple and continuous branding opportunities for an advertiser.

The interactivity around or within the commercial is database driven to fully maximize the server side capabilities of the control application processes. Users' responses can be gathered instantaneously, the effectiveness of a campaign e quickly measured, and changes made to the campaign to make maximize effectiveness. For example, an advertiser may want to offer a specified number of free samples of a new product recently launched. The control application can wrap that offer around a commercial, with "Special Value" (hyperlink) buttons presented, to encourage the user to fill out a sample request. After the specified number of requests have been received, the system can dynamically change the interactivity by either canceling the offer

altogether or replacing the offer with something else. A set of customizable hyperlinks may be provided in conjunction with the video window, to activate applets, scripts, "daughter windows" positioned for an unobstructed view of the stream, or perform other known user interface functions.

Ad insertion is a process in which one or more advertisements are inserted in one or more archived video streams or live video broadcasts (Program). The Ad Insertion Platform, according to an embodiment of the invention, seamlessly inserts audio/video advertisements (or gifs /flash ads) before, during or after archived video content, eliminating the need to host the ad and video on the same server. The central servers host the ads, and intelligently insert, or control insertion of them, into the presented video stream. This eliminates current technology limitations of encoding and storing the advertisements and streams on the same platform.

The present method provides the benefit of no buffering delays between the ad and video content. Additionally, the Ad Insertion Platform according to the present invention provides narrowband users with a higher quality viewing experience with the actual video content, by providing additional time for loading of the featured video content, while the inserted ad is playing. Thus, in instances where the ads are cached locally (or provided on distributed media), the presentation time for the ads may be used to prebuffer media content for later presentation. Thus, the actual media delivery rate can be slower than real-time, since the advertisement display time is available for download. Further, the appearance of a cached advertisement may be triggered, at least in part, by a state of the media buffer, which indicates an impending shortfall. Preferably, in such a case, an advertisement may be inserted at the next preencoded insertion point, to provide time for the buffer system to return to a normal state.

Presently, for an advertiser to present a video advertisement in conjunction with content, a copy of the advertisement must be hosted, and streamed from the same server as the content, because the advertisements must be encoded along with the video stream as one single stream and stored on the same streaming server as the video stream itself. As a result, the same advertisement has to be shown to everyone viewing a particular video stream, thus severely limiting targeting capabilities. If the advertisement and content are not within the same stream, known systems require considerable buffering delays (10-30 seconds) between advertisement and the video stream(s). There are no interactivity features built into existing ad insertion vehicles.

The system and method according to the present invention address these problems. The present invention thus provides a single location for ads, and allows for the insertion of an ad in any third party stream. This eliminates current technology limitations of encoding and storing

the advertisements and streams on the same platform. Because ads are inserted from a centralized location, it can offer advertisers a single point of contact for inserting the same advertisement in multiple streams that are hosted with multiple content providers. The present invention can insert advertisements of any format into live or on-demand video streams. For example, a web publisher is serving a Quicktime (.mov) movie to its user base. The system has the ability to insert an ad that is of any format: Quicktime (.mov.), Windows media (.asf, .wma), or RealPlayer (.rm). Since the system does not perform the play list functionality, it has the ability to call both the commercial and the existing stream at the same time. Once the commercial starts playing, the Insertion mechanism buffers the existing stream at the same time. Once the commercial has ended, the existing stream plays immediately. Thus, there are no intervening buffering delays. The present invention can insert different ads within the same video stream, depending on the user specifications. The system can also target based on the origin of the video request. Since the present invention does not insert ads in the stream, but rather inserts a unicast (commercials) stream into a multicast (video content) stream, the unicast stream may be customized based on user information.

The present invention also allows for the insertion of an ad in a live stream being broadcast by any third party. With other models, broadcasters and/or content providers must buy enabling hardware or move their broadcast to the ad serving provider's server farm. The present invention provides a distributed software solution, whereby web publishers simply need to communicate limited information with the central servers. One type of information communicated is when the commercials are to be inserted. Using control application software, a broadcast producer can submit commercial break information ahead of time or instantaneously. For example, an advertiser might seek to present advertisements Weekly or Daily: A producer can submit information about its weekly programming schedule (For example, Monday 9:30 AM to 4:30 PM, commercial breaks at 10, 30 and 50 minutes past the hour). A producer can also use an interactive screen to send information to the central server for processing further, to trigger an instantaneous commercial. For example, a commercial break is coming up in 5 seconds, 10 seconds, 30 seconds etc. The broadcast producer also enter information about the length of the commercial break (30 seconds, 60 seconds, 90 seconds, etc.)

Combined with the functionality of the "Interactive Screens" and the "Ad insertion Utility", the system can seamlessly insert advertisements into any live broadcast, not requiring any additional hardware or encoding software; the broadcaster or content provider continues to manage their broadcasts from their current facility. The present invention provides the broadcaster with a seamless and efficient ad insertion process.

The ad insertion utility is used to uniquely identify each user and manage the ad insertion workflow process. According to a preferred embodiment, each user is prompted to download the utility, a small program executing on the user's computer or media browser, the first time that the user attempts to watch a video steam through the system. Each time a user clicks on a video stream from a cooperative site, the utility sends to the central server, the site and section they are coming from. At this time the user is categorized and placed into a channel based on their profile and the web site they are accessing. This allows the system to send different users different ads (when they are accessing the same site or the same content from different locations) based on their channel placement. The ad insertion utility also controls the media stream presentations.

The Broadcast Utility allows Broadcasters to communicate with the central servers in real-time. The main purpose of this utility is for broadcaster to notify the central servers when commercial breaks will occur in their live broadcast. Integration of the ad-serving technology platform into a website is as simple as embedding a module of Javascript and HTML code in the web page source code.

Through an Active-X control (in case of Internet Explorer or AOL browser), or a Java applet control (in case of Netscape browser), the client (i.e., user system) preferably maintains a list of all the cached commercials, so that when a particular commercial needs to be shown to the user, its presence is checked in the cached commercial list, and if it is present, shown from the cache. The commercials may be identified specifically, or as a class. For example, commercials in different languages may be present, and automatically presented based on client-defined parameters. If the requested commercial is not present, then it is queued for download in proper sequence, or an animated gif for that commercial may be shown to the user, and a hint is provided to the download utility to download the commercial for later presentation. In case the user has a browser other than Internet Explorer, AOL, or Netscape and for non-Windows platforms, animated GIFs may be shown to the user, permitting cross-platform compatibility.

A first aspect of the present invention provides a system and method for seamlessly inserting a rich media advertisement into online video content, in a manner that maintains coordination and control of presentation while eliminating the need to host the media and advertising content on the same server. Therefore, a content media server system can be both physically and logically separated from an advertising media server, yet provide a seamless integrated experience for the user. Therefore, the various types of content may be in independently determined formats, for example archived formats or live formats of the same or different kinds. In particular, the advertisement can be either presented live or in real time, or prestored on a server, e.g., in

compressed format. According to the present invention, ads may also be stored local to the user, the presentation of which is controlled in conjunction with the media content.

The systems and methods according to the present invention are compatible with interactive media presentation systems, and in particular an advertisement may be delivered with an interactive application that provides additional information to the user and collects data from the user when provided. An application may also support electronic commerce, for example allowing the user to define or make a purchase.

A particular aspect of the invention provides a system and method that allows independent digital media streams to be presented to the viewer in continuous segments, without buffering delays between segments. Therefore, an interactive or otherwise non-predefined presentation may be provided without apparent discontinuities. Particularly, the media content may be interspersed with advertisements in a continuously presented stream, without presentation buffering delays.

A preferred embodiment of the invention performs this seamless presentation as follows. As is known, media playing applications require a small buffer of a media stream in local memory, in order to accommodate temporary degradation in stream bandwidth, and to ensure that an entire block of data is available for decompression and processing. Preferably, the system according to the present invention employs a standard third party presentation application, such as Realplayer, Quicktime or Windows Media Player. These known applications, as noted above, require prebuffering, but also are able to process only as single media stream, and thus when changing streams, introduce an intervening prebuffering latency due to the requirement of treating each stream as a separate session. In contrast, the present invention provides an application wrapper for the media presentation program, interacting at an API or event level. The application notably is capable of launching and controlling multiple media player sessions simultaneously, although typically only one session will be presenting media at any given time.

The first session launched must introduce a prebuffering latency, as is required by the underlying media player application. However, as the player application nears the end of its presentation, its communication bandwidth requirement abates, resulting in available communications channel bandwidth. The player application notifies the application that the downloading has ceased, and the channel bandwidth is therefore available. The application then immediately, before the first session has completed playing its media presentation from the buffered portion, commences prebuffering the subsequent session. Therefore, after the initial prebuffering period, a continuous buffered stream may be presented to a viewer, though the use

of multiple media player sessions under common control.

These various media player sessions are independent, and indeed may draw their respective media content from different servers. Thus, this technique allows the entertainment media content to be separately hosted from the advertising content. The present application further provides a means for communicating between independent media player sessions, allowing cross interactivity without requiring the media streams themselves to be interactive.

According to the present invention, a set of control parameters, which may be predetermined, adaptive, or dynamically generated, are provided to control the application program. Typically, this set of control parameters is established based on an analysis of the various media streams as well as a set of rules, requirements or preferences for the presentation.

According to a preferred embodiment, the various sessions are managed by a remote server, which communicates parameters with the application program periodically, e.g., every 2 seconds. However, in certain circumstances, a more autonomous presentation is supported, in which the application manages and controls itself. Preferably, in the case of advertising subsidy of entertainment media, the application provides a reliable and secure report of at least aggregate media and advertising consumption data for auditing and compensation accounting. Therefore, in one embodiment, the present invention preferably supports anonymous reporting of confirmation, for audit purposes. This anonymity preserves user privacy for at least two concerns; first, the user's explicit selection of media and direct interaction with the system, and second, an implicit profile generated of the user based on monitoring of activities over time and comparison with other users.

Thus, the present invention is capable of delivering advertising interspersed with online video content, from separate and/or separately controlled sources, without buffering delays between the advertisement and the featured video content.

According to a preferred embodiment of the invention, an Ad Insertion System is provided within an ad-serving platform, which is designed to enable intelligent insertion of commercials at the beginning, during, and after video content. The system achieves the intelligent insertion of commercials regardless of the connection speed. The system also allows caching of commercials on the user's hard drive in order to minimize the requirement for streaming on narrowband connections. The Ad Insertion System consists of two components: an Ad Insertion Control, and a Caching utility. The intelligence to insert a commercial is dynamically distributed throughout the process flow in the system, and interaction between different processes occurs in multiple ways. The decision to deliver a particular commercial in a particular format is not linear nor does it follow a single thread. Rather, a parallel, multithreaded

process is employed. The technology seeks to use the path of least resistance to arrive at a particular commercial to be shown. Should this path not be available, the system uses a cascade down approach to arrive at an equivalent decision. In a preferred embodiment, the intelligence is ever-evolving and adaptive, depending on the user's media player type or media player version. The system is therefore capable of adapting to play the right media format.

Typically, the structure of a media presentation, for example, insertion points or potential insertion points for commercials or objects appearing on-screen, will be provided in a metadata or out-of-band communication. Thus, a parameter file may be associated with the content. This parameter file is then processed to determine a set of rules or actions, which may be coextensive or differing from the media content parameter file, and indeed may represent data from multiple parameter files, for example for each segment of the presentation or for a plurality of interactive segments. The processed parameter file is then employed by the control application. The parameter file itself need not be totally predetermined, and may be provided or processed in small portions, and if processed remotely, delivered to the control application over a computer network before any action must be taken, allowing a dynamic, adaptive, interactive and/or variable presentation to be made, potentially responding to viewer and/or advertiser and/or media content owner considerations. The parameter files and processed parameter files are typically processed and stored in a central Internet-server database system associated with the control application, which generally also will serve as advertising server or control intelligence for serving advertisements.

The control application preferably encompasses a database, allowing a set of interactive buttons can be customized and dynamically updated for each advertiser. These buttons encourage immediate interaction with the brand and provide additional value to the users. Advertisers can use these interactive buttons to facilitate trial via a coupon or sample, provide more information, generate semi-qualified sales leads or facilitate e-commerce transactions. Users benefit from the ability to immediately obtain valuable information on products of interest as well as from special values offered by the advertiser. Users' interactions with the buttons are collected and classified in the database for performance reporting, aggregate learnings and future targeting efforts. The database may reside at a remote server, local to the user, or have local and remote portions. Preferably, the control application has a local database, and periodically communicates pertinent parameters with a remote central database, such that normal control application activities can be handled from locally stored data. However, certain interactive and dynamic media presentations may require on-line handshaking and interactive communications.

The present invention thus supports a business model that targets websites and streaming

video providers as clients, who will in turn sell the capabilities of the platform to their advertisers. The provider of the platform will therefore be able to generate revenues by offering a complete ad-serving technology platform to websites and/or streaming video providers, earning a 1) per client partnership fee and 2) revenue share or per ad served fee. This scheme, in turn, allows the platform to become universal, since it is not integrally associated with any particular content provider or advertiser. Of course, other business models are supported in which a closed system provides an advertiser or content owner an ability to exercise control over the user(s).

The system can deliver an audio/video advertisement (or any other media type) embedded in any location of a web page; or in a pop-up window that can be launched on any web page; or inserted at the beginning, end, or during an archived video stream; or inserted at the beginning, the end, or during a live video broadcast in absolute or relative time. The platform can deliver a full screen or less than full screen audio/video advertisement to a user's TV screen, if the TV is connected to a set-top box with a static or dynamic network address, an operating system and a media player. The platform allows advertisers to include interactivity as part of the advertisement, which can be encoded as a part of the video presentation (e.g., hyperlinks to portions of the image at a given time) or using screen buttons, generated by the control application, surrounding the video window. The advertisements can be targeted, based on a user's profile, ad viewing, and surfing (e.g., hyperlink usage) behavior. The system can track the user's ad viewing behavior from the first time they enter a cooperative web page. Using a robust data capture schema, every ad view and button click made by the user is captured in a database.

The platform may be completely database driven and contains several databases. These databases are divided into two categories: Media Databases and Psychographic Databases. The technology delivers an advertisement based on the user's preferences and other user related information and the web site information that is sent back to central databases. This information is preferably captured through JavaScript scripts that are customized to capture user profile information (player type, connection speed, browser type, user id-if the user has viewed a system ad before) and the web site information (Web Site Name, Web Site Section Id, Commercial ID). Once the system has gathered this information, the technology sends the user the appropriate advertisement and provides a set of customized screen buttons appropriate for the circumstances. Since the Platform is completely driven by database functionality, any aspect of it may be customized "on the fly."

According to one embodiment, the present invention provides an incentive program for

encouraging users to accept, and even seek out, multimedia advertising. This incentive program has primarily two foci; first, the compensation to the user is preferably directed toward subsidy of the broadband connectivity to the deliver medium, e.g. the Internet, and second, the incentive is awarded only on a positive feedback from the user that the user was present during playback of the commercial. Thus, an embodiment of the invention addresses the limitations of the prior art in two ways; it promotes broadband connectivity to the delivery medium, thus reducing the burden of the transmission bandwidth requirement on the user, and assures the advertiser that the user did, in fact, view the commercial. As an alternative to a customized feedback button, each commercial may terminate with an alphanumeric code, which transiently appears on the user screen. The user is then required to communicate the code, by electronic communication, for example through a web page, applet, e-mail or otherwise.

It is possible, according to the present invention, to provide targeted commercials without a priori knowledge of the viewer's identity. This can be done in two ways. First, the user system may define a client-side filter, to extract commercials from a multicast, dependent on the user profile. The system may then account for these commercials, both to the advertiser and to the user. The targeting may also be performed by using an external resource, for example Doubleclick or BeFree, to identify a commercial to be targeted based on the service provider identification of the user and the relevant characteristics thereof. In this case, compensating the user for viewing the commercial requires rather a collaboration with the service provider, or a dual set of relationships. If anonymity of user profile is a concern, then the user accounting may include a set of messages that transmit in a first message, using a secure digital signature, that a particular user has viewed an unidentified commercial, optionally placing a value on that commercial or aggregate value on a series of transactions. A second secure anonymous message is transmitted identifying the commercial, but providing no information identifying the viewer. Alternately, a pure anonymous transaction may be possible, in which the payment to the user takes for form of a token, entitlement to which is established by watching the commercial. Therefore, when the commercial viewing is confirmed, the user system, for example using a secure anonymous challenge-response authentication protocol, receives the token. Correspondingly, the advertiser's account is debited for the transaction. Other secure anonymous communications techniques are known; see, Hassler, Vesna, et al. "MiMi: a Java Implementation of the MicroMint Scheme"; Chomicki, Jan, et al., "Decentralized Micropayment Consoliation"; Azbel, Ilan, "Payword Micro-Payment Scheme. Strengths, Weaknesses and Proposed Improvements."; Neumann, B. Clifford, "Security, Payment and Privacy for Network Commerce", IEEE Journal, 13(8), Oct. 1995; Mellare, Mihir, et al., "iKP – A Family of Secure

Electronic Payjment Protocols" (July 12, 1995); Chaum, D., "Achieving Electronic Privacy", Scientific American (August 1992) pp. 96-101; Chaum, D. et al., "Untraceable Electronic Cash" Proc. Crypto '88; Brands, Stefan, "Untraceable Off-line Cash in Wallets with Observers", each of which is expressly incorporated herein by reference.

The computer software application according to a preferred embodiment may also be modified and applied to other uses. For example, the software can be customized and co-branded for channel marketing, corporate communications, Continuing Practice and Education (CPE), employee training and other peripheral uses, based on its ability to present multimedia information in an interactive fashion, with network communications capability. In an educational system, the interactive process and confirmation of viewing may be used to implement a protocol, for example comprehension testing. The interactivity therefore allows a merging of the presentation and testing, within a single application framework, which preferably provides a high degree of integration with the Internet and web browser. If a user does not pass a test, a prior course segment may be repeated, or supplemental materials presented. After the segment is successfully completed, a subsequent unit may be presented.

It is therefore an object according to the present invention to provide a system and method for delivering media streams, in which streams are organized in a multithreaded manner to permit simultaneous presentation of a first stream and buffering of a second stream.

It is also an object of the present invention to provide a system and method for presenting a targeted advertisement, comprising classifying a consumer, for example by assigning to the consumer a behavior classification based on a set of expressly defined set of preferences, an automatically derived set of preferences, and a prior purchasing behavior; selecting a targeted advertisement to be delivered to the consumer, based on the classification; presenting the targeted advertisement to the consumer; and monitoring a user action immediately subsequent to the presentation of the targeted advertisement. According to another embodiment, a system and method are provided for delivering a multimedia advertisement to a consumer, comprising means for performing the steps of selecting an advertisement to be delivered to the consumer; presenting the advertisement to the consumer; and requiring an input by the consumer immediately subsequent to the presentation of the advertisement to confirm attention thereto.

It is also an object of the present invention to provide a system and method for caching a targeted rich media, online advertisement, comprising the steps of providing a uniquely user with a caching utility; classifying the user by assigning a classification based upon a set of expressly defined preferences or behaviors, an automatically derived set of preferences or

preferences, and/or a prior purchasing behavior; selecting a targeted advertisement to be cached on the user's local system based on the classification; caching the targeted advertisement and expunging advertisement from the local system based on time, frequency or change in consumer classification parameters.

These and other objects will be apparent from a review of the drawings and detailed description of the preferred embodiments. This task is solved by the characteristics of the independent claims. Additional embodiments arise from the sub-claims.

BRIEF DESCRIPTION OF THE DRAWINGS

The preferred embodiments of the invention will now be described by way of the drawings, in which:

Fig. 1 shows a sign-in screen;

Fig. 2 shows an application window having a set of iconic advertising buttons, as well as application control and interaction buttons; and

Fig. 3 shows the application window of Fig. 2, with a media player window;

Fig. 4 is a block diagram of a system according to the present invention, wherein the central server passes information through an advertiser server to a user;

Fig. 5 is a block diagram of a system according to the present invention, wherein the central server passes information independently of an advertiser server to a user;

Fig. 6 shows a flow diagram representing key processes of a method according to the present invention;

Fig. 7 shows an advertisement insertion system block diagram according to the prior art;

Fig. 8 shows a block diagram of an advertisement insertion system according to the present invention;

Fig. 9 shows a block diagram of an ad insertion control application according to the present invention;

Fig. 10 shows a process and example of an economic model for a business exploiting the system and method according to the present invention;

Figs. 11 and 12 show a popup player with and without wallpaper, respectively, according to the present invention;

Figs. 13 and 14 show an ad-insertion player with and without wallpaper, respectively, according to the present invention; and

Figs. 15 and 16 show a schematic diagrams of a digital rights management infrastructure according to the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

EXAMPLE 1

As shown in Fig. 1, a user is requested to identify (username) and verify (password) himself on use of the system with a sign-in screen; Fig. 2 shows an application window having a set of iconic advertising buttons, as well as application control and interaction buttons. On clicking an advertiser icon button, a video media ad is called up and presented to the user.

As shown in Fig. 3, a media player is invoked, having a video window and interaction screen buttons. To stop the ad, the close button is selected. After the ad is finished playing the user may click "OK", to receive the reward, or "Instant win", as described below. Likewise, other functions may be presented. The media player window is "always on top", and therefore cannot readily be hidden during presentation. The application window also provides various management and portal functions, and of course may present updated news, weather, financial information, and the like, as is well known in the art.

When the Ad is playing, only the "Close" button is active, and the other buttons are inactive. After the commercial is over, the other 3 "action" buttons get activated for about 8 seconds; if the user clicks on one of the buttons, appropriate and corresponding action will take place; If the user does not click on any button, after 8 seconds, the "action" 3 buttons become deactivated again and the commercial starts all over again.

According to one embodiment of the invention, an online video ad may be converted into graphic interchange format (GIF89) advertisement, upon the completion of the video advertisement. Thus, after the advertisement is presented, an image may persist as a banner ad (for example with a hypertext link), allowing delayed interaction by the user without occupying a multimedia session.

According to a preferred embodiment of the invention, a software suite is provided which is distributed across a server and a set of clients. The server software interacts with client software for accounting, to provide control parameters, and to provide fresh content. The client software provides an advertising preference and management function (including communications with the server) as well as a commercial viewer. The advertising preference and account management function application is called ZeBar™. The ad viewer software is called ZeView™. The client software suite preferably performs the following functions: Loads ZeBar™ as the viewer machine boots up; shows an ad in a variable size Viewing Window with sound and motion; stores activity locally in the desktop and remotely on the web (every activity performed by a user related to ZeBar™ and ZeView™). This includes, but is not limited to, recording the following: Did the user close ZeBar™; Clicking on a button to see an ad; What

was the location of the button on ZeBar when it was clicked; Which button was clicked; Was the person online or offline; What ad was played; Was the ad completed; Did the user click on the OK button on time; Did the user visit the web site (see detail on "More Info" Button on ZeView™); Did the user click on the system proprietor's logo to visit the system proprietor's web site; and Did the user click Instant Win button; the software recognizes the connection speed or lack thereof and as such do the following: If the connection speed is 128 Kbps or less, prompt user to insert the CD in the CDROM drive; If the connection is missing, prompt user to insert the CD in the CDROM drive; If the connection speed is more than 128 kbps do not prompt the user to insert the CD in the CDROM drive; and the ZeView™ closes automatically if the user opens a new application.

The ZeBar™ is how the user views the ads and commercials, manages his profile(s), manages his point bank, and interacts with the system proprietor. ZeBar™ is designed in such a way that a user finds it non-intrusive, minimally annoying and as a source of information based on which he/she will make intelligent purchasing decisions. At no time does ZeBar™ take priority over any other activity a user may want to perform. A user is, for example, allowed to position the bar along any edge (left, right, top or bottom), anchor at one place, or hide it.

The preferred ZeBar™ embodiment has buttons that are 30X60 pixels for the side and 24X60 for the top and the bottom position. Each button has a graphic interchange file (GIF) graphic imposed on it. This GIF is preferably always uniform in size shape and colors. There are preferably no more than 16 buttons on the ZeBar™. Out of these 16 buttons, 4 buttons are reserved and cannot be used for any particular advertiser. (see details on Reserved Buttons). The ZeBar™ has an indication for a user, that by clicking here he/she can manage his account. The ZeBar™ always has the user name on it, to ensure that the right user account is credited for the user activity. Any user has the opportunity to change the user login, by clicking on a button or some type of link. An Internet cookie or similar means is employed to automatically identify the user of the client system to the server. Clicking on any button starts a process that depends on the button clicked: If a user clicks on an Ad button, the process initiated comprises the following: checking the connection, or a lack thereof, and the connection speed, and spawn another process described earlier (the CD Rom Process), if the connection speed is insufficient; start the preferred media player; and display an advertisement in the ZeView™. If the user clicks on any of the reserved buttons, they spawn independent processes.

The ZeView™ window is the part of the screen where the ad is displayed and user interaction with the ad is processed. This space shall be no more than 468X468 pixels and is divided into two sections: The AdView Section, which is the space that will be used to provide

all the commercials. This space is not be more than 468X468 pixels. A customized version of QuickTime™ plays the steaming media or the media read from a local device depending upon the connection, its speed or lack of any connection. This customized version is a slightly modified version which, for example, does not ask user any questions regarding updates to QuickTime player and does not encourage user to buy new versions of QuickTime or to visit any sites related to QuickTime.

The User Interaction Section is used to provide buttons (30X60 in size) for the user to indicate his preferences: The button on the right is the button that a user will need to indicate that he completed an ad viewing transaction. This button has "OK" imprinted on it. The button has a permanent position on the User Interaction section. This button remains inactive while the ad is playing. At some instance before an ad finishes, this button becomes active for an amount of time that is, for example, randomly determined. This amount of time is no less than 5 seconds but no more than 8 seconds.

The button on the center has the inscription "Instant Win". This button spawns an "Instant Win Process", that hyperlinks to a referenced Internet web site, and will be activated only in the following situations: If the user is connected to the Internet, and If the user is over 18 years old and If the user resides in a state where such games are permitted by law. The "Instant Win Game" button is instantly deactivated if the user presses the OK button. This button becomes activated for exactly the same amount of time as the OK button, described above. The user has an "either or" choice between pressing the OK button and INSTANT WIN button. Should the user press the INSTANT WIN, he forfeits any credit that he would have earned for watching the ad. In essence, the user is buying a chance to play the INSTANT WIN game by paying in credits. Once the user presses the INSTANT WIN button, a process residing in a referenced Internet web site, which randomly decides on a prize. This prize has a value range from 0 points to a point equivalent of $100. The expected (average) value of any prize is exactly equal to the credit a user would have earned had he pressed the OK button, according to the formula: (p(WIN)*DollarAmountof WIN=Credit). The process, at completion, opens a window, announcing the outcome of process. Should the user win a prize, the window gives directions on how to claim that prize. A user may be limited to no more than 5 Instant Win chances per day, or some other number or limit. The button will not thereafter be deactivated, but rather will function as an OKAY button after it has been temporarily exhausted.

The third button has the inscription "More Info". This button, when pressed, initiates processes as follows: If the commercial/ad is about a service or a durable product, the process initiates a daughter window stating that by pressing another button in the daughter window, the

user is agreeing to send his name/address/contact info to the advertiser of the product/service. If the commercial is about a low value product (food, household chemicals, and other "supplies"), the process initiates a window offering the user to print a "cents or dollars off" coupons and provide useful tips on the product. Should the user choose to print the coupon, he or she may again press another button in the daughter windows agreeing to the terms of use. All these activities are be recorded and transmitted to the central server at an appropriate time. Should the user be connected to the Internet or a connection made, the information may be transmitted instantaneously. Should the user be off line, the information is stored in a format that is tamper resistant, e.g., using encryption techniques. This information is then transmitted to the central server at an appropriate time. Advertisers may be offered an option to customize the page that will be called by the More Info button. Advertisers will be able to customize this pages based on the inventory of parameters offered.

The ZeBar™ has four reserved 30X60 buttons. The first button is called "Hit Me!" with a logo in the background. This button, when pressed, plays a totally random ad in the ZeView™ window. This button has the top most positioning on the ZeBar™ among the reserved buttons, but is always be positioned lower than the ad icons. The purpose of this button is two fold: To make all users become familiar with the system proprietor's logo embedded in a button. A logoed button will be the way the system proprietor will make its mark an all the content sites, free or otherwise, we will use the ZeBar™ to popularize the system proprietor's logo. The second purpose is to ensure that user curiosity and interest remains high. While it is expected that the sweepstakes and the Instant Win games will keep the interest high, the randomness of the process is used (as is done on TV) to maintain a high level of curiosity for the users. This is achieved by having this button play a totally random ad. This ad may or may not match a user profile. The "Hit Me!" button opens a dialog box that has two options: Play a random ad; and Play a random ad periodically, for example every 4-60 minutes, as defined by the user. Should the user input a number within the given parameters, the client software corrects the error and stores the information appropriately. This process should open a box with a timer on it and have a countdown of 10 seconds (i.e., a commercial will start in 10,9,8... seconds). The box allows users to stop (cancel) this process and close the window. If the user does not close the process, the AdView process begins at the end of the countdown process. When the Hit Me button is pressed, it will show a clip based on one of the icons that are present on the screen. Since the information is already present in a user computer, there will only be limited number of calls made to the server.

The second button has the inscription "Instant Win" on it. By pressing this button, the

user experience will be the best among those experienced by pressing any other button. This button is positioned below the last Ad button. This, button, wherever permitted by law, when pressed once, plays a totally random ad in the ZeView™; deactivates any Instant win button on the ZeView™ area; at the end of the ad, activates the OK button on the User Interaction Section; should the user press the OK button without timing out, initiate an "Instant Win Process" on the proprietor's internet web site; once the Instant Win Process is complete, displays the results in a daughter window; does not give any credit to the user for watching an ad; and closes the ZeView™ window once the transaction finishes (either the user presses the OK button or the OK button times out).

The third button has "NOW!" inscribed on it. Pressing this button will allow user to look at daily specials for lunch in his area; any sales that may be going on in geographically proximate regions, as determined by zip codes; allow any coupons that may be available to print and to redeem; and presents information on new releases/musicals etc. Pressing this button will dynamically generate a page that will have all this information custom tailored for a user profile.

The ZeBar™ also has a button with a piggy bank GIF on it. Clicking on this will simply show how many points a user has in his account. The user uses this button to interact with the central server site. Pressing this button takes the user to that section of the site where he can: login; change his profile/address and other information; check his point bank; buy points; redeem points; and ask for more info.

In addition to targeting the user based on a user profile, the present invention may also present advertisements (static or multimedia) based on geographic location and/or time. For example, local restaurant lunch specials, time sensitive coupons, and the like, may be presented in this manner. Thus, for example, a user registration process defines a zip code, which allows the user location to be specified. Therefore, local advertising content may be inserted or presented. This is presented to the user, for example, through a "Surprise" button, which provides an opportunity for geographically and chronologically targeted advertisements, which may also be targeted to the user. For example, lunch discounts at a local restaurant are displayed midday, while in the evening, other offers are made. This potentially allows an advertiser to, on short notice, and for a short duration, promote specials based on an actual load condition

The "MORE INFO" button, described above, may be employed, to perform advertiser-specified actions, for example to present a questionnaire, survey, application form, etc., or to complete the business transaction (purchase movie ticket, make a dinner reservation, or purchase merchandise viewed in the ad). A user is preferably incentivized by further "credits" or discounts, for complying with information requests or

otherwise accepting additional advertising content. The "More Info" button sends a message to the server, which will generate required information in HTML, which will open up a browser on the client system for viewing and interaction. The text on the button may be changed dynamically depending on the requirement of the advertiser.

Multiple users with access to the same computer may select and maintain different profiles. Generally, the accounting is consolidated, although different accounts may be maintained for different users. Each user may have a separate login, with optionally stored password.

According to one embodiment of the invention, if a multimedia advertisement is not to be integrated with another multimedia content presentation, a plurality of choices may be provided for interaction after a linear commercial is presented. First, a simple acknowledgement, indicated by "OK", allows the user to directly return to normal activities. Second, the user may request more information ("MORE INFO") regarding the sponsor of the message, for example by opening an Internet web page for the sponsor. Third, an alternative interactive option, such as "INSTANT WIN" is provided to allow the user to continue an interaction with the system, while digressing from the immediate past commercial sponsor. In this case, a contest or sweepstakes may be provided, to hold the user's attention, while he is presented with other advertising information or the like. Such advertisements may be automatically or manually triggered. In an automatic triggered embodiment, a timer or latency-based system is provided to periodically present a commercial. For example, if a user is logged on for one hour per day, and four commercials watched per day are required for complete subsidy of broadband access, then commercials might automatically be presented every fifteen minutes. Likewise, a user profile may define the occurrence parameters for the commercials. Alternately or additionally, commercials may be triggered manually. For example, if the user wants to take a break from other activities, he might trigger an ad, which defines, for example, a one-minute rest period. This trigger can be, for example, in the form of a "HIT ME" button. The commercial or ad presented may be randomly selected, or defined based on a user profile, user preference, or sponsor targeting or broadcast requirements.

EXAMPLE 2

According to a second embodiment of the invention, a streaming media system is provided which allows multiple streams to be processed concurrently, with coordinated management. Key processes of the system are represented schematically in Fig. 6.

As a part of a cooperative scheme, a web publisher integrates a set of HTML/Java

script/Java code (hereinafter called the "Z-Scripts") into every page on its web site where a commercial message has to be inserted. This code serves several purposes. It defines the size of the commercial window; the location on a web page of the commercial window; the type of the commercial message (pop-up/embedded); the site and section from which the call for commercial is made to the central server; the behavior of the frame (where the commercial is played); the location of the central server; and the beginning and the ending media type, identification and location of such media. This script thus communicates with central servers, and manages the work flow process, providing information on user capabilities and making appropriate requests for the right commercial message based on profiling and user machine capabilities.

As a user arrives at a cooperative web page, the web site makes a request for a commercial message. The central servers look for a cookie or other stored information that was collected previously. If no cookie is found, the central servers initiate a server-based process. This process checks: The user connection speed (56k or broadband); the user browser type (Internet explorer or Netscape); and availability of a media player on a user's machine. (Windows Media Player, Real player and or QuickTime player). If the cookie is not present, the servers leave a cookie on the user machine. Depending on the connection speed and player type, the servers initiate the following process: If the connection speed is over 128kbps and a media player is present, the servers stream the advertisement directly from the server; if the connection speed is over 128kbps and a media player is not present, a Java Applet is downloaded. This applet allows an advertisement to be played on a website even in the absence of a player; if the connection speed is less than 128 kbps and a media player is present, the user is requested to download a proprietary plug in (hereinafter called "Z-Utility"). If the user agrees to download Z-Utility, the utility is downloaded and installed automatically. If the user does not agree to download Z-utility, the user is shown an animated gif.

After the advertisement is delivered, the Z-Scripts on the web page allow the system to record an impression for billing and auditing purposes. It also covers the black space on the web page where the advertisement was placed, thus creating another advertising impression, and allows the system to record the site/section and other relevant information for profiling and targeting purposes. Should a user act on the interactive buttons, the middleware tracks the location of the button on the player and all information entered into any request forms. For example, shipping information entered in the HTML window in response to a special value button for "Free Sample" is returned to the server by the notification Servlet for onward transmission to the fulfillment house.

The Z-Utility is a piece of software that is designed for users with narrowband connections (56k or less). It allows these users to watch good quality (Similar to a Broadband experience) video/audio advertisements. The Z-utility works as follows. When a narrowband user comes to a cooperative site, the user is prompted to download the Z-Utility. (Each user only has to download the Z-Utility once, when they first encounter a cooperative web page). The Z-Utility is installed on the user's machine and is executed every time a user connects to the Internet. The utility is in constant communication with assigned central servers. This assignment can be dynamically changed should it be required. The utility is sent a 'hint' every time a user goes to a cooperative web site. The utility immediately calls the assigned central server, and makes a request for an advertisement, based on the following information: The site id (a unique number used to identify a specific Web publisher.); The section id (a unique number used to identify a specific Web publisher's webpage.); the advertisement name; the type of presentation (pop-up or embedded); the format type (.asf, .rm, or .mov); and how many times a particular advertisement can be shown to a user. The Z-Utility then starts downloading the advertisement while keeping a record on how many times an advertisement can be shown and has been shown. When the user comes back to the same site, if the advertisement is already cached in their machine, it is played immediately. At the end of the advertisement, the notification is sent to the server and the counter is incremented by 1 in the Z-utility database. The count is also incremented in the local machine. As soon as the counter reaches the maximum impressions for a particular advertisement for a particular user, the advertisement is deleted from the user's hard drive and additional advertisements are requested. If the advertisement is not presented, the information is returned to the central server, an animated gif is delivered. Further ads are then downloaded until all appropriates ads are cached on the user's machine.

The data capture process for embedded and pop-up ads proceeds as follows. When an advertisement is served to a user's machine, a notification is sent to the central server which records: Advertisement (campaign id; a campaign is set of media served to individual user that includes: properly formatted media file (.mov, .rm, .asp, .fla), button ad .gif, Z-Player, wallpaper .gifs, Special Value button .gifs and Special Value button redirects, if necessary); web site on which it was seen; section on the web site on which it was seen; user's IP address; date/time; any special value button clicks; whether the user has a narrowband connection; and whether the utility has been downloaded.

Advertisements can be inserted at the beginning and/or at the end of a video clip; at a relative time (based on a time relative of the played program), e.g., at 1 minute after the

beginning of the program); back to back in a block of advertisements; and at an absolute time, e.g., advertisements which are played at a absolute time, for example, at 3:30PM.

The Ad insertion on-demand process occurs when a user wants to view archived video content on a web publisher's web site. The process starts when a user clicks on the archived video content (Assume the user already downloaded the ad insertion utility). At this point, the utility sends the user's info (Cookie info) in order for the system to identify the user. The central database places the user in a channel in order to serve them additional advertisements during the process of the video stream. The central database servers also receive the Site ID, section ID and the location of the archived content. Based on the User and Site information, the central servers serve the appropriate ad to the user. At the same time, the utility calls the appropriate content from the web publisher's server farm in order to start the buffering process while the user is watching the advertisement. As soon as the ad is complete the content is played. If it is desired by the web publisher to include advertisements during the playing of an archived stream, the central server serves the user ads at the relative times the publisher has set, based on the selected channel.

The system may insert ads before, during, or after video content with no buffering delay between the ad and video. The video itself is therefore in a paused state, while the ad or insertion is presented. The communication stream including the video may continue, and therefore the buffering and pausing allow an alteration in the delay between download and presentation. Known so-called trick play techniques may also be applied to the buffered data.

According to a preferred method, each web publisher continues to manage and host its own content, without having to send each updated piece to a 3rd party to re-cut, re-digitize and reprocess. The central server hosts the ads from its servers and intelligently inserts them (or controls insertion) into the video stream. The ad insertion platform also provides buffer time for loading of the featured content, which allows narrowband users to have a higher quality viewing experience of the feature video content as well.

The ad insertion process is executed at the user's computer. If the user has not downloaded the Ad Insertion Control, The user is prompted to Download the Ad Insertion Control (Active X control). The architecture of this application is shown schematically in Fig. 9. The functionality of the Ad Insertion Control is distributed in two components: A player engine, which is an intelligent process that operates on the player, and a "dumb" player. The player engine provides an interface to the player and Servlets residing on the Central Server. It is responsible to get the commercials information from Central Server and intelligently parses it and then according to the commercial parameters, instructs the player to intelligently play the

commercials. The "dumb" player is a component that provides the basic player functionality. This component of Ad Insertion Control actually plays the commercials and other video contents on instructions provided by the Player Engine

The user chooses to download the Ad Insertion Control by clicking "YES" on the download prompt. A Servlet call (A piece of software that is located on Central server that communicates with a client's machine, i.e., to deliver properly formatted ad), is made that sends the following to the Central servers: site_id, section_id, channel_id (Unique number used to determine section level or user profile level targeting), presence of interactive layer flag; realtime streaming flag (RTP/RTSP/UDP/HTTP), program URL, program frame URL, media format type (RT/WMP/QT), and video content URLs for different connection speeds (up to 6). The Servlet generates a HTML (or other) page according to the above specifications and then serves the appropriate combination. The ad Insertion System is installed on the user's local storage drive. The System consists of the following: the Ad Insertion Control; the Caching utility; and a Media player and connection speed detection functionality (Checks for media player and type). The Servlet returns a Commercial Insertion Profile String to the Ad Insertion Control that consists of: commercial URLs, commercial frame, ad insertion parameters (timing), special value identifiers, and special value action URLs. The Ad Insertion Control receives, reads, and parses the commercial insertion profile string.

The Control organizes the commercials in three buckets, each of which represents the following: Bucket 1: Commercials Played Prior to Video Content Delivery; Bucket 2: Commercials Played During the Video Content Delivery; Bucket 3: Commercials Played After the Video Content Delivery. For each bucket, the Ad Insertion Control takes out one commercial URL and the corresponding commercial frame URL from a particular bucket. The Commercial frame is displayed in the video content window until the commercial starts playing. The Ad Insertion Control then checks to checks to see if the commercial is available on user's hard drive. If the commercial is available on the user's hard drive, Ad Insertion Control begins playing commercial from the user's hard drive. If the commercial is not found on the user's hard drive, then the commercial is streamed directly from the central servers after buffering enough data. The Control hides the commercial frame before starting to play the commercial. The Ad Insertion Control sends a notification to the Web publisher's HTML page that the commercial has started and instructs it to refresh the special values buttons and wallpaper (if needed). If the requested commercial was not available on the user's hard drive, a hint is dropped on the user's hard drive informing the Caching Utility to start downloading the commercial that is currently being streamed and all other commercials in the respective ad

category. The Caching utility starts the caching process, thus politely caching the commercial on the user's local storage drive for later use. The Ad Insertion Control calls a Servlet to notify that a commercial has been served.

For bucket 1 (commercial played prior to media stream presentation), after the Ad Insertion Control checks for, and determines that the commercial is not available on the user's local storage drive, it begins to buffer the commercial. During the buffering period, the Control displays the commercial frame. Thus, the user must wait for the first commercial to be served. If there are multiple commercials in the bucket and while the Control is playing the commercial, it starts the buffering process (in the background) at t-10 seconds in order to prepare the next commercial in the bucket to play. For bucket 2 (commercial played interspersed with media presentation), while the Control is playing the requested video content, the Control starts the buffering process (in the background) at t-10 seconds in order to prepare the commercial to play. For bucket 3 (commercial played after stream presentation), when the Ad Insertion Control intelligently detects that the video content (being played) is ending in next ten seconds, it starts buffering the first commercial from this bucket in order to provide the seamless transition from video content to commercial.

The user may decline to download the Ad Insertion System by clicking "No" on the download prompt. If the system loses control of the user's session, it informs the Web Publisher that this has occurred.

The following Tables A, B and C detail the operational parameters of the three common presentations of the Control Application: A. Embedded in a web page; B. Pop Up on the user's screen, and C. Ad Insertion within a media stream. The Pop-up application is also shown in Figs. 11 and 12, showing, respectively, the pop-up application with and without wallpaper, and the ad-insertion player is shown in Figs. 13 and 14, showing, respectively, the pop-up application with and without wallpaper. When embedded in a web page, the control application according to the present invention provides only a video window, with user interface elements defined by the host web page.

Table D shows a process diagram for the establishment of an advertising campaign using the system and method according to the present invention. Table E shows a process diagram for the serving of a statically defined ad to a viewer using the system and method according to the present invention. Table F shows a process diagram for the serving of an interactive ad to a viewer using the system and method according to the present invention. Table G shows a process diagram for interacting with a user who has selected a "Special Value" interactive ad using the system and method according to the present invention.

Table A. Embedded in Web Page

Item	Description
Size (h x w)	Recommended: 228x210 (Total Player), 140x184 (Video Window) Minimum: 220x156 (Total Player), 116x128 (Video Window)
Recommended ad placement	In the top right position on the page (the right column if the page layout is based on three column design)
Video stream format compatibility	• Windows media player 6.4 and above • Real Player G2, 7 and above • QuickTime 3 and above • Flash 4 and above
Browser Compatibility	• Internet Explorer 4.0 and higher • Netscape 4.0 and higher
Interactive buttons	• Min 1 button/ Max 3 buttons • Total number of characters for interactive buttons are as follows: - For Recommended size, 24 characters total (approx. 8 characters/button) - For minimum size, 15 characters total (approx. 5 characters/button) • Information behind buttons can be hyperlinks or custom creative.
Player Controls	None
Commercial Delivery:	• Narrowband users who accept to download the Z-Utility and all broadband users will experience the following commercial experience with interactive buttons: - Commercial may be proceeded by non-streaming media (animated gif/gif) - Commercial streamed from server(s) if the connection is broadband. Else, the commercial is played from the user machine if the commercial is present on the hard drive. If the commercial is not present on the hard drive, animated gif is delivered. - Commercial is followed by non-streaming media (animated gif/gif) • Narrowband users who decline to download plug-in will see animated gif with interactive buttons only.
Download Utility	• The download of the Z-Utility is only prompted for narrowband users. The Z-Utility allows the system to politely and intelligently cache audio/video ads on a user's hard drive, so that the ad can be immediately recalled once a user arrives at a cooperative Web page. • The Z-Utility only needs to be downloaded once for both pop-up and embedded ads and works across all cooperative sites. • Size of the Z-Utility is 115 kbs. • Once the Z-Utility is installed, it is executed every time a user connects to the Internet. The utility downloads and caches commercial.
Impression Counting Process	• Impression counted as 1 video ad once a commercial begins to stream (first frame received by the I-frame) • Impression counted as 1 gif ad if the commercial cannot be streamed for any reason and animated gifs is played. • Impression counted as 1 Flash Ad once a Flash commercial starts playing
Codecs used	• Windows media player (video: MPEG4V3, audio: Clep.net) • QuickTime Player (video: Sorenson 2.1, audio: IMA4) • Real Player (video: Real, audio: Real)
FPS	10-15 FPS
Size of ad unit	15kb/second (225Kbto 450Kb)
HTML placement	Just below the header. Will not affect page load as the code is I-frame based
Sample of HTML code	<IFRAME id="Zframe" name="Zframe" SCROLLING=no FRAMEBORDER=0 WIDTH=212 HEIGHT="228"

	SRC="http://ad1.zebus.com/bbs/CHECK_SPD?site_id=1&category_id=1&render_type=e&run_time=28&reload_gif=1"> <ilayer id=Zframei width=100 height=300 visibility=hide></ilayer> </IFRAME>

	<LAYER id=Zframe width="100" height="300" visibility="hide"
	clip="0,0"
	src="http://ad1.zebus.com/bbs/CHECK_SPD?site_id=1&category_id=1&render_type=e&run_time=28&reload_gif=1&browser=N"
	onload="moveToAbsolute(Zframei.pageX, Zframei.pageY);visibility='show';clip.width='300';clip.height='300'" z-index="0" right="0" left="0">
	</LAYER>
	<script> if (document.layers) { var features1 =
	'width=1,left=1,top=1,height=1,directories=no,location=no,menubar=no,scrollbars=no,status=no,toolbar=no,resizable=no';
	check_win =
	window.open("http://ad1.zebus.com/bbs/SET_COOKIES",'check',features1);
	self.focus(); if(!check_win.opener) check_win.opener= self;
	}
	</script>

Table B. Pop-Up

Item	Description
Recommended Size	Total Player Video Window 500(w) x 350(h) 210(w) x 165(h) or 320(w) x 240(h) (if Wallpaper is desired).
Recommended placement	In the top left position on the page
Video stream format compatibility	• Windows media player 6.4 and above • Real Player G2, 7 and above • QuickTime 3 and above • Flash 4 and above
Browser Compatibility	• Internet Explorer 4.0 and higher • Netscape 4.0 and higher
Interactive buttons	• Maximum of 6 buttons • Maximum number of characters for each interactive button is 30 characters (2 rows of 15 characters). • Information behind buttons can be hyperlinks or custom creative.
Player controls	Close button in upper right corner of window (on the Windows caption bar)
Commercial Delivery:	• Narrowband users who accept to download plug-in and broadband users will experience the following commercial experience with interactive buttons: – Commercial streamed from server(s) if the connection is deemed to be broadband. Else, the commercial is played from the user machine if the commercial is present on the hard drive. If the commercial is neither presented on the hard drive, animated gif is delivered. • Narrowband users who decline to download plug-in will see animated gif with interactive buttons only.
Download Utility	Works same as embedded. If user already downloaded for an embedded, they do not need to download it again.
Impression Counting Process	• Impression counted as 1 video ad once a commercial begins to stream (first frame received by the I-frame) • Impression counted as 1 gif ad if the commercial cannot be streamed for any reason and animated gifs is played. • Impression counted as 1 Flash Ad once a Flash commercial starts playing
Codecs used	• Windows Media Player (video: MPEG4V3, audio: Clep.net) • QuickTime Player (video: Sorenson 2.1, audio: IMA4) • Real Player (video: Real, audio: Real)
FPS	10-15 FPS
Size of ad unit	15-30kb/second (225Kbto 900Kb)
HTML Placement	Just below the header. Will not affect page load as the code is I-frame based

Sample of HTML code	<SCRIPT language=javascript type=text/javascript>
	function MM_openBrWindow(theURL,winName) { features =
	'width=500,height=340,left=10,top=235'
	x = window.open(theURL,winName,features);
	if (x.opener == null) x.opener = self;
	}
	</SCRIPT>
	<BODY
	onload="javascript:MM_openBrWindow('http://ad1.zebus.com/bbs/CHECK_SPD?site_id=6&category_id=1&render_type=p','PlayerWindow')" >

Missing at the time of publication

Table D

Key Process Point	Data Input	Client Integration
Execution	1. Define client as Web publisher or agency 2. Assign campaign parameters (site_id, section_id, and channel_id) 3. For ad insertion, input ad insertion system source code 4. Create campaign and Z-Player 5. Gain approval from client on campaign parameters and creative	1. Web publisher inputs 5-10 lines of HTML/Java script source code onto every Web publisher webpage (includes initialization Servlet call to retrieve the campaign information (site_id, section_id, and channel_id)) where a commercial message would be shown
Technical Result	1. Web publisher or agency inputted in to central database 2. Campaign in database grouped into one ad category, defined by site_id, section_id, and channel_id	1. Central server can serve a campaign through one (or many) Web publisher's page
Business Result	1. Central server or servers can control and modify campaign parameters from server side	2. Central server tracks and charges client on a CPM and per special value click basis
Key Criteria for Possible Outcome #1	• Central server defines channel_id = 0	• Central server includes Ad Insertion System source code
Possible Outcome #1	• Campaign will be targeted based on section level profile (i.e., a "travel page" on a Web publisher site)	• Campaign will be served within a video stream (ad insertion)
Possible Outcome #2	• Central server defines channel_id > 0	• Central server does not includes Ad Insertion System source code
Possible Outcome #2	• Campaign will be targeted based on user level profile (i.e., men 25-50 yrs. old throughout website)	• Campaign will be served via embedded or pop up platform only
Exception Handling	• Central server defines campaign parameters incorrectly, database constraints will not accept input. Additionally, upon campaign parameter input, database will edit check the site_id, section_id, channel_id at relationship level and will log errors (email message).	• Client modifies or incorrectly incorporates source code on webpage, code will raise an exception and a Servlet sends an error to the log file (triggers email). If site_id is able to be read correctly, a pre-defined Web publisher default campaign will be served until issue is rectified. If site_id is not able to be read by the servers, a pre-determined default .gif (needs to be determined) will be served until the issue is rectified. • Client requests modification of source code, client will follow the change process: fill out a change request form, gain approval, test.

Table E

Key Process Point	Identification of User Campaign	Identification of User Capabilities
Execution	1. Central server server receives ad call with site_id, section_id, and channel_id and other attributes 2. Central server drops cookie on user machine for 1) reporting, 2) future profiling, 3) ability to send hint to Central server, and 4) any other purposes	1. Central servers send and communicate (multiple back and forth Servlet calls) with user's machine to determine connection speed, media player type, browser type and other key attributes that may be needed to effectively deliver a commercial message
Technical Result	1. Central server knows what campaign to serve	- Central server knows and classifies user speed, player type (and browser type) - If connection speed is => 128kbs, user classified as broadband. - If connection speed is < 128kbs, user classified as narrowband. Narrowband users are prompted to download Caching Utility. - Central server can serve properly formatted ad to user
Business Result	1. Central server can serve consumer a targeted campaign	1. Central server serves optimal media based on user capabilities
Key Criteria for Possible Outcome #1	• User machine and Central server are both online	• User machine and Central server are both online
Possible Outcome #1	• Central server server receives ad call	• Central server communicates with user machine
Key Criteria for Possible Outcome #2	• User machine and Central server disconnect during online session	• User machine and Central server disconnect during online session
Possible Outcome #27	• Central server receives ad call upon user's next visit to webpage	• Central server communicates with user machine from beginning upon next user webpage visit
Exception Handling	- If site_id, section_id or channel_id cannot be read by Central servers, an exception will be raised and a Servlet sends an error to the log file (triggers email). If site_id is able to be read correctly, a pre-defined Web publisher default campaign will be served until issue is rectified. If site_id is not able to be read by the central servers, a pre-determined default .gif (needs to be determined) will be served until the issue is rectified.	- If user and central server connection becomes disabled or disconnected, the central server will time out and will re-send a Servlet in the next online session. - If the central server does not receive a Yes answer (or any answer) to any of its queries during the communication process, it will serve a .gif file.

Google Exhibit No. 1008
Page 311

Table F

Key Process Point	Interactive Platform: Serve Campaign	
Execution	1. Server sends properly formatted media campaign to user machine that may include: a streaming commercial (HTTP/UDP/RTSP/RTP), Z-Player, Wallpaper and Special Value button creative/URLs, .gif 2. Notify server when commercial, and/or .gif has been shown 3. Notify server of user's media player and render type (i.e., what type interactive ad)	
Technical Result	1. Central server can serve a campaign to user	
Business Result	1. Central server can serve and verify campaign delivery to client	
Key Criteria for Possible Outcome #1	• BB (Broadbnand) connection with MP (Media Player)	
Possible Outcome #1	2. Stream commercial with correct commercial file 3. Notify server when ad has completed and show .gif	
Key Criteria for Possible Outcome #2	• BB connection without MP (Java-enabled)	
Possible Outcome #2	2. Stream a movie that can be decoded and delivered by a Java or other technology applet/application 3. Notify server when ad has completed and show .gif	
Key Criteria for Possible Outcome #3	• BB connection without MP (no Java)	
Possible Outcome #3	1. Show .gif (or other appropriate media format, i.e., Flash) 2. Notify server .gif (or other media) has been shown	
Key Criteria for Possible Outcome #4	• NB (Narrow Band) connection without Caching Utility, with MP	
Possible Outcome #4	1. Central server prompts user to download Caching Utility 2. If user accepts Caching Utility download: a) Show .gif or other appropriate media; b) Notify server .gif (or other media) has been shown c) Install Caching Utility (IE: Activex; Netscape/Mac: JavaApplet); d) Start caching ads on Caching Utility in proper media format e) Create index in Caching Utility of downloaded ads	3. If user declines Caching Utility download: a) Show .gif or other appropriate media b) Notify server .gif (or other media) has been shown c) Set cookie flag, do not prompt again in this session (defined as browser open)
Key Criteria for Possible Outcome #5	• NB connection without Caching Utility, without MP	
Possible Outcome #5	1. Central server prompts user to download Caching Utility 2. If user accepts Caching Utility download: a) Show .gif or other appropriate media	3. If user declines Caching Utility download: a) Show .gif or other appropriate media b) Notify server .gif (or other media) has been

	b) Notify server .gif (or other media) has been shown c) Install Caching Utility (IE: Activex; Netscape/Mac: JavaApplet) d) Start caching ads on Caching Utility in WMP default format e) Create index in Caching Utility of downloaded ads	shown c) Set cookie flag, do not prompt again in session (defined browser open)
Key Criteria for Possible Outcome #6	• NB connection with Caching Utility, with MP, with campaign cached	
Possible Outcome #6	1. Central server queries Caching Utility for appropriate ad category campaign 2. If appropriate campaign is present: a) Show commercial from a local storage drive b) Notify server that commercial shown upon completion and show .gif	
Key Criteria for Possible Outcome #7	• NB connection with Caching Utility, with MP, without campaign cached	
Possible Outcome #7	1. Central server queries Caching Utility for appropriate ad category campaign 2. If appropriate campaign is not present: a) Search Caching Utility for other campaign grouped in same ad category b) If other campaigns present, show commercial from a local storage drive and notify server c) If other campaign not present, show .gif and notify server .gif has been shown d) Caching Utility sends hint to central servers to download appropriate campaign and other ads in same ad category e) Central server begins to cache appropriate campaigns on Caching Utility	
Key Criteria for Possible Outcome #8	• NB connection with Caching Utility, without MP, without campaign cached	
Possible Outcome #8	1. Show .gif or other appropriate media 2. Notify server .gif (or other media) has been shown	
Key Criteria for Possible Outcome #9	• NB connection with Caching Utility, without MP, with campaign cached	
Possible Outcome #9	1. Show .gif or other appropriate media 2. Notify server .gif (or other media) has been shown	
Exception Handling	• If user disconnects with central server during Caching Utility download, user is re-prompted to download upon next visit. • If user disconnects with central server during caching process, any part of the commercial cached is purged, and the next time the user is online, the Caching Utility contacts the Servlet to request the commercials. • If user disconnects with central server during campaign delivery, complete campaign is delivered upon the user's next visit. • If requested commercial in cache has error, central server will time out after 60 seconds and then show a .gif file. • If only part of commercial is cached and user disconnects, the cache does not stay on Caching Utility. The next time user is online, the Caching Utility contacts the Servlet to request the commercials.	

Table G

Key Process Point	Z-Player Interaction (Special Value))	Data Processing	Database Reporting
Execution	1. Action URL redirects user and collects click through, IP address, time of click, site_id, section_id, redirect URL and other needed/suitable information 2. Central server collects user inputted information for HTML pages for: above listed info, mailing info and up to set of 15 attributes (text, list, radio buttons)	1. all user collected information written into flat .txt files (or other suitable format) 2. Data collected every hour (incremental info) 3. End of day, auto process collects flat .txt files from all web servers and all files put into database for aggregation and insertion into reporting tables	1. Client requests up to 4 online reports with proper security information (password) 2. Central database generates reports 3. Requested report is uploaded to www.Zebus.com for online viewing 4. System stores most updated reports in database archive as .xml flat file
Technical Result	1. System collects user information	1. System aggregates user information	1. System reports user info. (T+1 report)
Business Result	1. System can collect user metrics for reporting purposes	1. System can aggregate ads served and user metrics for reporting purposes	1. System can provide reporting to clients
Key Criteria for #1	• User clicks on any Special Value button	• Any client campaign is active (dates or impressions)	• System reporting interface is online/running
Possible Outcome #1	1. System database records and classifies user interaction based upon click through data	• System database calls web servers every hour to request transfer of any flat .txt files (or other pre-determined interval)	• System database server receives client request for online report • System database server requests and receives correct authentication information (password) and database servers generate and display requested client reports
Key Criteria for #2	• User does not click on any Special Value button	• No client campaigns are active	• System reporting interface is not online/running
Possible Outcome #2	1. System database records and classifies user visit with no interaction	• System database does not call web servers to request data transfer	• When System reporting interface down, database server notifies default server of day, time and length of time
Exception Handling	• If Central server is unable to redirect URL, user receives an error message due to the time out of the browser • If Central server is unable to collect user submitted info, user is re-prompted (if due to improper validation) or shown message that unable to accept and error noted to log file (If not due to improper validation) (email sent) • If server is disabled, load balancer will send to another server.	• If System database server cannot accept data on server disk, information will be recorded on redundant disk (same server – carbon copy) • If redundant disk is also disabled (with main server disk), Central server loses data and error is sent to log file. • If Central server is disabled, load balancer will send to another server. • If system master load balancer is disabled, redundant load balancer activated.	• If System report cannot be generated, client receives error notice and database server notified, error log filed. • If master database server disabled, slave database activated as master and email error log is filed. • How reporting handles data errors: Master database collects all data and sorts indefinable formats into error log.

EXAMPLE 3

Fig. 4 shows a block diagram of a system according to the present invention, wherein the central server passes information through an advertiser server to a user. Thus, according to this model, the viewer apparently interacts with a single entity, the advertiser, and the central server is shielded from direct communication. In this case, the advertising content is preferably not hosted on the central server, since this might result in unnecessary transmissions of the streaming media. This model corresponds to an interactive TV Ad insertion system.

As shown in Fig. 4, a user visits a partner's web page that is enabled with Z-Scripts (A). Z-Scripts are executed, delivering the user's connection speed, media player and browser type, and the partner's site information to the system servers. A matching process is performed between the User Profile (B) and Media (C) databases in order to determine the following: The Advertisement to send to the user, based on the site information and user profile; The type of Advertising Media (Video, .gif, Flash, Banner, etc.), based on the user's capabilities and advertising requirements; The type of advertising format to send, based on the user's media player type (Windows Media, RealPlayer, or Quicktime); and the corresponding special values.

The user is sent the Targeted Advertisement (D) with the corresponding Special Values (E) via the Embedded or Pop-up Z-Player (F). The user then has the option to interact with the Special Value Button (E). If the user clicks on a Special Value Button (E), then an event 6a is triggered and the corresponding Special Value Creative (G) is sent to them. The Special Value Creative (G) may request 6b that the user submit information (Brochure, Sample, etc.). Any submitted information is collected and classified by the central database.

The User Profile Database is where the system stores demographic (when provided) and non-personally identifiable behavior-based information. The Media Database is where the system stores all of the media files (Advertisements (D)), Special Value Buttons (E), and Special Value Creative (G).

The Special Value Buttons are used by advertisers to capture additional user information, provide users with samples or brochures, link users to Web pages, etc. The buttons are hosted in the Media Database (C) and linked with the corresponding Special Value Creative (G) and media file (Targeted Advertisement (D)). The Special Value Creative is simply the Sample or brochure form, linked web page, coupon, etc. that is presented to the user as a result of them clicking on the corresponding Special Value Button (E). The creative can be hosted either by the central server, the advertiser, or a third party.

Fig. 5 is a block diagram of a system according to the present invention, wherein the central server passes information independently of an advertiser server to a user. In this case,

the user's system is aware that the control application is communicating with a different server than that controlling the normal web pages from the web site. This architecture potentially allows the user greater privacy, since the particular web sites have only indirect access to the user profiles and the information on which they are based. This model corresponds to a web page ad insertion model.

According to this model, a user visits a partner's Web page and chooses to view a media file that is enabled with Z-Scripts (A). Z-Scripts are executed A, delivering the user's connection speed, media player type and browser type, and the partner's site and file information to the central servers. The Z-Scripts call the client's Content Database (I) in order to capture the appropriate media file (Client Content (H)). A matching process is performed between the User Profile (B) and Media databases (C) in order to determine the following: The Advertisement to send to the user, based on the site information and user profile; the type of Advertising Media (Video, .gif, Flash, Banner, etc.), based on the user's capabilities and advertising requirements; the type advertising format to send, based on the user's media player type (Windows Media, RealPlayer, or Quicktime); the corresponding special values; and the appropriate client media file to play after the advertisement.

The user is sent the Targeted Advertisement (D) with the corresponding Special Values (E) via the Ad Insertion Z-Player (F). After the advertisement is complete, the user is sent the desired client content (H). The user has the option to interact with the Special Value Buttons (E). If the user clicks on a Special Value Button (E), then an event is triggered and the corresponding Special Value Creative (G) is sent to them. The Special Value Creative (G) may request that the user submit information (Brochure, Sample, etc.). All submitted information is collected and classified by the central database.

The Client Content represents the media file that the user is selecting to view on the client's web site. The Client Content Database is located at the client's hosting facility. This is the database where they store all of their streaming content (H).

Fig. 7 shows a block diagram of an advertisement insertion system according to the prior art. In this scheme, it is particularly noted that presentation of each segment incurs a mandatory prebuffering latency in all of the known digital media players. Therefore, each advertisement or other change in media flow will cause a disruptive interruption. Since the known media player implementations are single tasking, a first stream must end playing, even if it is contained entirely within a local cache, before a subsequent stream can begin setup.

In contrast, Fig. 8 shows a block diagram of an advertisement insertion system according to the present invention. According to the present invention, the control application provides

multithreaded support for simultaneously handling multiple media streams. Thus, when a first media stream is completely downloaded, and thus the communications bandwidth is available, a subsequent media stream may commence prebuffering, even during completion of playback of a prior stream. Likewise, the present invention allows locally cached media, such as commercials, to be interspersed with downloaded streaming media, such that the actual presentation data rate for the media may be in excess of the available communications bandwidth. Thus, when the buffer for the downloaded media is depleted, a locally cached commercial may be presented while the buffer is regained.

Fig. 10 shows a process and example of an economic model for a business exploiting the system and method according to the present invention. As shown in Fig. 10, a business process is provided in which advertisements for Internet web sites may be separated from the content delivery. Therefore, the present invention allows servers to be established for hosting streaming advertisement content, which may be employed without a disruption of a continuous media presentation. The advertising control central server therefore need not host the advertisements in order to provide effective management and control over the process. The advertising control central server also typically deals with or through web sites (publishers) to satisfy their technical requirements, rather than serving advertisers directly. Thus, advertisers will retain their relationships with the publishers. Fig. 10 also shows that the advertising control central server seeks only a small portion of the advertising revenue, preserving the advertisement driven publishing model.

In some cases, a live broadcast or webcast is provided, in which it is desired to provide targeted commercials or customized segments. Thus, while most of the presentation is common for all viewers, other portions are individualized. Therefore, according to one embodiment, when a user seeks to join a webcast, the user is identified, for example, with username and password. The server then assigns the user a channel (i.e., both a physical communication layer and a predefined protocol) for advertisements, which are also broadcast, but which need not be broadcast in real time, especially where the user's communications bandwidth exceeds the amount necessary for the webcast. Therefore, in order to reduce peak load on the server and communications link, commercials may be precommunicated, in whole or in part, to users, and triggered from a local cache at the appropriate time. Before a commercial is about to begin, the sever may transmit a signal indicating the start time. At the designated time, the user system accesses the channel, which may be live or buffered, and presents the commercial. At the end of the commercial, the system then continues to present the webcast stream. Thus, the system according to the present invention may provide a seamless integration of digital video streams

from multiple sources, by coordinating their respective presentation parameters prior to playback, to ensure that each stream is cued for immediate commencement of playback when the prior stream completes.

EXAMPLE 4

Figs. 15 and 16 show schematic configurations of a digital rights management (DRM) architecture according to the present invention. The caching utility provides a basis for managing the download and storage of stream buffer files on the user's hard disk. While it is preferably to persistently store Ads in a convenient format, it is also preferred to limit access to content outside the control of the player. This preserves the value of the content for the owner. According to the present invention, as the proprietary content is streamed to the user's computer, the caching utility may exert control over it. For example, even if the media player software or content stream format does not support DRM, this may be implemented by limiting access to and use of the cached data. Thus, by limiting the size of the cache, and expiring and deleting old data, abuse of the temporarily stored data will be limited. Likewise, the caching utility may encrypt (or scramble) data as it enters the cache, and decrypt (or descramble) it as it is recalled by the media player.

As shown in Fig. 15, the player software (engine) encapsulates the application programming interfaces (APIs) for the media player, providing a generic interface to a virtual player. A middle tier layer provides targeting and profiling interface to the engine, transaction processing capabilities, and reporting. The player software thus provides an opportunity to add functions to the media player software.

Fig. 16 shows that, by restricting the playing of controlled content by the virtual player, DRM rules may be stored and enforced by the central server. To play content subject to DRM rules, there is a two step handshake process. The Zplayer, preferably through a SSL connection, talks to the central server. It sends the players unique global user identification (GUID) and requests a UserAgent. The central server creates a new UserAgent, and adds an entry in one of the tables for the GUID, IP address, and a expiry date about 30-60 seconds into the future. The ZPlayer makes the actual call for the media, which is a servlet call. The Zebus servlet checks the UserAgent and the time in the table. If the UserAgent matches one in the table and if the time is not greater than the Expiry date, the media is sent, otherwise the request is rejected. The requesting user IP address may also be checked. It is noted that, in some cases, communication with the central server may be conducted periodically, wherein the player operates autonomously in accordance with preset rules between communications.

This prevents a user from trying the URL directly in a media player, because the actual media URL is not directly visible. Even if the user can find the actual URL, if the UserAgent does not match, then the request is rejected. An important part of this is the two-stage handshake protocol and the 30 sec window of validity of the UserAgent. Even if the user can circumvent the UserAgent, the 30 sec expiry of the UserAgent gives the user a very small window to directly open the URL in a media player directly.

The present invention also supports pay-per-view schemes. Since the content is protected for playback except through use of the Zplayer and associated systems, and the Zplayer is controlled through a central server, a content owner may be assured that compensation will be received for use of content. Thus, the content may be streamed to the user, or stored or buffered locally in a protected format and played only when authorized. This authorization may entail an accounting event. The present invention may also be integrated with a traditional entertainment or video system. For example, the system could be used to substitute for commercials broadcast on a television channel. The user, for example, is compensated for viewing the commercial from the system of the present invention, instead of from the existing source. All processing can be handled locally. The present invention may also be integrated with a digital video recording device (DVR), potentially allowing the broadcaster to implement the commercial substitutions. In this case, the broadcaster will be able to increase revenues by providing targeted advertisements, and by providing interactive opportunities, including potentially a full electronic commerce portal. Thus, as another aspect of the invention, digital multimedia streams may be integrated with live or analog multimedia streams.

Aspects of the present invention may also be applied to non-PC Internet devices, such as PDA/wireless devices and TV set-top devices. Therefore, it is understood that the present invention is not limited to commonly understood broadband technologies for delivering commercial quality (or better) video to users through a copper-backbone Internet. Rather, the present invention provides the ability to serve users various types of advertisements, hosted under unrestricted conditions, in a manner suitable for presentation on a user's system. The architecture therefore provides two components: a central server for coordinating and accounting for usage of the user's, content owner's and advertiser's systems, and a client software suite, hosted on the client hardware, for communicating with the central server and presenting the content and advertising to the user.

Many aspects of the present invention employ known techniques, although employed in different contexts herein. Therefore, it is understood that these known and techniques and those

associated with them may be employed in conjunction with the present invention, to the extent consistent therewith.

It is also understood that the various aspects of the invention may be employed together, individually or in subcombination. Further, it is understood that the present techniques are not limited to use on the Internet, as presently known, and may be applied to a large number of human computer interface systems.

While the above detailed description has shown, described and pointed out the fundamental novel features of the invention as applied to various embodiments, it will be understood that various omissions and substitutions and changes in the form and details of the system and method illustrated may be made by those skilled in the art, without departing from the spirit of the invention. Consequently, the full scope of the invention should be ascertained by the appended claims.

CLAIMS

What is claimed is:

1. A method for delivering a targeted advertisement, comprising the steps of:

classifying a consumer by assigning to the consumer a behavior classification based on a set of expressly defined set of preferences, an automatically derived set of preferences, and a prior purchasing behavior;

selecting a targeted advertisement to be delivered to the consumer, based on the classification;

presenting the targeted advertisement to the consumer; and

monitoring an action immediately subsequent to the presentation of the targeted advertisement.

2. The method according to claim 1, wherein the monitored action is employed to update the expressly defined set of preferences.

3. The method according to claim 1, wherein the monitored action is employed to update the automatically derived set of preferences.

4. The method according to claim 1, wherein the targeted advertisement is a rich media presentation.

5. The method according to claim 1, wherein a purchasing behavior is monitored to update the automatically derived set of preferences.

6. The method according to claim 1, wherein the targeted advertisement is selected to control a purchasing behavior.

7. The method according to claim 1, wherein the presenting and classifying are performed on a computer system local to the consumer.

8. The method according to claim 1, wherein the classifying is performed on a computer system remote from the consumer.

9. The method according to claim 1, further comprising the step of requiring an input by the consumer immediately subsequent to the presentation of the targeted advertisement to confirm attention thereto.

10. The method according to claim 9, further comprising the steps of receiving a payment from an advertiser for the targeted advertisement; and guarantying a viewing of the targeted advertisement.

11. The method according to claim 1, further comprising the step of receiving compensation from an advertiser for display of advertisements based on a cost per verified impression.

12. The method according to claim 1, wherein an available but unselected potential action is employed to update the automatically derived set of preferences.

13. The method according to claim 1, wherein the consumer is presented with a plurality of options after presentment of the targeted advertisement, and a user selection of a one of the plurality of options is employed to update the automatically derived set of preferences.

14. The method according to claim 1, further comprising the step of incentivizing the consumer to receive the advertisement.

15. The method according to claim 14, wherein said incentivizing comprises implying a monetary value for receipt of the advertisement.

16. The method according to claim 14, wherein said incentivizing comprises providing a discount for goods or services to the consumer.

17. The method according to claim 1, further comprising the step of outputting a message to the consumer after delivery of the advertisement.

18. The method according to claim 1, further comprising the step of printing an output message to the consumer after delivery of the advertisement.

19. The method according to claim 18, wherein the printed output message comprises a coupon having a unique identifier thereon.

20. The method according to claim 1, wherein said monitoring step further comprises requiring the user to input a context-sensitive message.

21. The method according to claim 1, wherein said monitoring step further comprises requiring the user to input a message selectively defined by the targeted advertisement.

22. A method for delivering a multimedia advertisement to a consumer, comprising the steps of:

selecting an advertisement to be delivered to the consumer;

presenting the advertisement to the consumer; and

requiring an input by the consumer immediately subsequent to the presentation of the advertisement to confirm attention thereto.

23. A method for publishing advertisements, comprising the step of receiving compensation by a publisher from an advertiser for display of advertisements based on a cost per verified impression.

24. An content presentation method, comprising the steps of:

providing a database containing a plurality of addressable media content;

providing an interactive application for displaying a selected media content from the database to a user;

verifying a viewing of the entire content by the user; and

providing a reward to the user for viewing the content.

25. The method according to claim 24, wherein the content comprises educational material.

26. The method according to claim 24, wherein the content comprises marketing material.

27. The method according to claim 24, wherein the content comprises a commercial.

28. The method according to claim 24, wherein the database is local to the user.

29. The method according to claim 24, wherein the content is remote from the user.

30. The method according to claim 24, wherein the application receives the content from the database through a switched packet network.

31. The method according to claim 30, wherein the switched packet network comprises the Internet.

32. The method according to claim 24, wherein the selection of media content is based on a profile of the user.

33. The method according to claim 24, wherein the selection of media content is based on a classification of the user.

34. The method according to claim 24, wherein the selection of media content is defined remotely from the user.

35. The method according to claim 24, wherein the selection of media content is defined locally to the user.

36. The method according to claim 24, wherein the application has a plurality of media content selections available for display simultaneously.

37. The method according to claim 24, wherein the interactive application requiring a manipulation of an object on a graphic user interface for said verification.

38. The method according to claim 24, wherein the interactive application senses a physical presence of the user in proximity to a display for said verification.

39. The method according to claim 24, wherein the interactive application provides a plurality of interactive opportunities after displaying the media content.

40. The method according to claim 24, wherein the interactive application display selected media content commencing periodically, with gaps therebetween.

41. The method according to claim 24, further comprising the step of accounting to a provider of the media content for verified display thereof to the user.

42. The method according to claim 41, wherein the accounting to the provider

maintains an anonymity of the user.

43. The method according to claim 24, further comprising the steps of providing a central server and a plurality of content providers, communicating between the application and the central server, and communicating between the central server and the plurality of content providers.

44. The method according to claim 43, further comprising the step of accounting for a verified viewing of the media content by a user.

45. The method according to claim 44, wherein the accounting comprises accounting, at the central server, to modify an account of the user and an account of a content provider.

46. The method according to claim 44, wherein the modification of the account of the user and the modification of the account of the content provider are of opposite and different magnitudes, with a difference of value of the absolute values comprising a profit for the central server.

47. The method according to claim 24, wherein the application and the media content database are connected through a communication channel capable of streaming media content in real time, and wherein the media comprises video data.

48. The method according to claim 47, wherein the application and the media content database are connected through a communication channel having a bandwidth in excess of that required to stream a single media content database record in real time, and wherein the media comprises video data.

49. The method according to claim 47, wherein the reward comprises a subsidy of a cost for maintaining the communication channel.

50. The method according to claim 24, further comprising the steps of identifying the user; and selecting media content in dependence on a user identification.

51. The method according to claim 50, wherein the user identification is associated with a user profile, the media content being selected according to an algorithm operating on user profile information.

52. The method according to claim 51, further comprising the step of recording a transactional history of the user, and providing the user profile based on the transactional history.

53. The method according to claim 51, wherein the user profile comprises demographic information.

54. The method according to claim 51, wherein the user profile comprises geographical information.

55. The method according to claim 51, wherein the user profile is adaptively updated based on data obtained subsequent to said selecting.

56. The method according to claim 51, wherein the user profile is maintained locally to the user, and defines filter parameters for locally selecting media content.

57. The method according to claim 51, wherein the media content selection is based on a correlation of a predetermined target profile for the media content and the user profile.

58. The method according to claim 57, wherein the selection is further based on a temporal sequencing algorithm.

59. The method according to claim 24, wherein the selecting is based on a valuation of a verified viewing by the user with respect to an available media content.

60. The method according to claim 24, wherein the database of media content is distributed across a plurality of servers, a selecting being based on a resource locator generated by the user.

61. The method according to claim 24, wherein the selecting is based on an Internet cookie stored locally to the user.

62. The method according to claim 24, wherein the verifying step comprises transmitting an encrypted message from the application.

63. The method according to claim 24, wherein the verifying step comprises transmitting a pair of secure messages from the application, a first message identifying the user and a second anonymous message identifying the selected media content.

64. The method according to claim 24, further comprising the step of soliciting responses from the user, and rewarding the user for providing responses to the solicitations.

65. The method according to claim 24, wherein the reward comprises a discount for a future purchase by the user.

66. The method according to claim 24, wherein the reward comprises a sweepstakes entry.

67. The method according to claim 24, wherein the reward comprises patronage loyalty incentive credits.

68. The method according to claim 24, further comprising the step of statistically accounting to a media content provider for a predicted number of verified viewings.

69. The method according to claim 24, wherein the media content is selected based on a express user request.

70. The method according to claim 24, further comprising the steps of providing, with the application, a set of selection choices for the user; and receiving a selection of a media

content choice from the user.

71. The method according to claim 70, further comprising the step of automatically changing a set of selection choices periodically.

72. The method according to claim 70, further comprising the step of permitting the user to manually change a set of selection choices.

73. The method according to claim 24, further comprising the step of limiting rewards available to a user within a predetermined temporal interval.

74. The method according to claim 24, further comprising the step of limiting rewards available to a user in at least two different predetermined temporal intervals varying in scale.

75. The method according to claim 24, further comprising the steps of maintaining a record of verified viewings, and performing an analysis of the recorded verified viewings.

76. The method according to claim 24, further comprising the step of recording instances of said providing step not followed by said verifying step.

77. The method according to claim 24, wherein said verifying comprises requiring a user input within a predetermined period of time after the end of a media content presentation.

78. The method according to claim 24, further comprising the step of presenting a common media content, wherein the common media content is interspersed with selected media content.

79. The method according to claim 78, wherein the common media content has predetermined insertion points for the selected media content.

80. The method according to claim 78, wherein the common media content is broadcast in real time, and where the application selects from a predetermined set of media content associated with the common media content.

81. The method according to claim 24, wherein said verifying step comprises performing a volitional action by the user which is prompted by the application and which varies on repeated instances of presentation by the application.

82. The method according to claim 24, wherein the verifying comprises a secure encrypted challenge response authentication sequence.

83. The method according to claim 24, wherein the application selects a common media content stream from a plurality of choices, and intersperses selected media content with the common media content.

84. The method according to claim 24, wherein a verification signal is transmitted by the application using a same communication channel as that used for receiving the selected

media content.

85. The method according to claim 24, wherein a verification signal is transmitted by the application using a different communication channel from that used for receiving the selected media content.

86. The method according to claim 24, wherein a said verifying further comprises printing a secure verification certificate.

87. The method according to claim 24, further comprising the step of providing a programmable media content presentation platform having a communications link with the Internet and executing an Internet browser application, wherein the application comprises a browser dependent applet.

88. The method according to claim 24, further comprising the step of providing a programmable media content presentation platform having a digital communications link, wherein the application comprises a software application which executes in an operating system environment.

89. The method according to claim 24, further comprising the step of providing a programmable media content presentation platform having a graphic user interface, wherein the application comprises a persistent desktop object.

90. The method according to claim 24, wherein the selected media content is MPEG-4 compliant.

91. The method according to claim 24, wherein the media content is selected based on associated MPEG-7 compliant metadata.

92. The method according to claim 24, further comprising the steps of providing a common media content, having an associated cost, and interspersing the selected media content with the common media content, wherein the reward is employed to offset the associated cost of the common media content.

93. The method according to claim 24, wherein said verification does not associate an identity of the user with a selection of media content.

94. A method for providing a seamless integration of a plurality of streaming media segments, comprising the steps of:

(a) commencing presentation of a first segment;

(b) determining a desired start time for a second segment;

(c) prebuffering the second segment during playback of the first segment with sufficient time for the second segment to be available prior to completion of the presentation of the first segment; and

(d) commencing presentation of the second segment seamlessly with a termination of presentation of the first segment.

95. A streaming media presentation control application, comprising:

(a) means for controlling a plurality of instances of media presentation software;

(b) receiving a message from a first instance of the media presentation software; and

(c) using the received message to control a second instance of the media presentation,

wherein only one of said first and second instances performs a media presentation at a time.

96. A system for seamlessly inserting a rich media advertisement into online video content while maintaining coordination and control of presentation, comprising:

(a) a first host for serving first media content;

(b) a second host for serving second media content;

(c) a control server, having a database of control parameters defining temporal aspects of said first media content; and

(d) a control application, receiving said first media content, said second media content, and an appropriate set of control parameters from said control server,

wherein said control server operates to seamlessly present said first and second media content sequentially without requiring a gap therebetween.

97. The system according to claim 96, wherein said first media content is an archival format.

98. The system according to claim 96, wherein said first media content is an real time presentation.

99. The system according to claim 96, wherein said control server performs a financial accounting function for presentation of at least one of said first and second media content.

100. The system according to claim 96, wherein one of said first and second media content is coupled to a financial subsidy for the other.

101. The system according to claim 96, wherein said control application provides a user-interactive control over presentation of at least one of said first and second media content.

102. The system according to claim 96, wherein said appropriate set of control parameters is dynamically generated by said control server.

103. The system according to claim 96, wherein said appropriate set of control parameters is adaptively generated based on past interactions with a user of said control

application.

104. The system according to claim 96, wherein said control application selectively provides a hyperlink.

105. The system according to claim 96, wherein said control application senses a change in communication status with at least one of said first and second hosts.

106. The system according to claim 96, wherein said control application simultaneously controls at least two media players.

107. The system according to claim 105, wherein said control application is adapted to pass messages between said at least two media players.

108. The system according to claim 96, wherein said control application communicates a status of presentation to said control server.

109. The system according to claim 96, wherein at least one of said first and second hosts is local to said control application.

110. The system according to claim 96, wherein said appropriate set of control parameters controls a telecommunications buffer status of at least one of said first and second media content.

111. A content presentation method, comprising the steps of:

providing a database containing a plurality of addressable media content;

selecting the advertisement to be delivered based upon defined user and advertisement classification parameters;

providing an online advertisement served from an advertisement server into a video stream, webpage or pop-up window;

displaying the selected advertisement to the user, followed by the video content, using an interactive application; and

collecting and classifying any user data provided.

112. A method for caching a targeted rich media, online advertisement, comprising the steps of:

providing a user with a caching utility;

uniquely identifying a user based on the utility;

classifying a consumer by assigning to the consumer a classification based upon a set of expressly defined preferences or behaviors, an automatically derived set of preferences or preferences, and/or a prior purchasing behavior;

selecting a targeted advertisement to be cached on the user's local system based on the classification; and

caching the targeted advertisement and expunging advertisement from the local system based on time, frequency or change in consumer classification parameters.

113. A method for managing content rights, comprising the steps of:

(a) receiving, by a user machine, content in a streaming format;

(b) buffering a portion of the content;

(c) presenting content from said buffer, wherein said buffering is controlled by a caching utility independent of a content presentation utility; and

(d) controlling the caching utility to limit receipt or storage of content based on a set of predefined content rights.

114. The method according to claim 113, wherein the caching utility is controlled to limit receipt of content based on the set of predefined content rights.

115. The method according to claim 114, wherein the caching utility is controlled to limit buffering of content based on the set of predefined content rights.

116. The method according to claim 115, wherein the caching utility is adapted to simultaneously interact with a plurality of content presentation utilities.

117. The method according to claim 116, wherein each content presentation utility is capable of managing only one content stream at a time, and wherein the caching utility in combination with a plurality of content presentation utilities is capable of simultaneously managing a plurality of content streams.

118. The method according to claim 113, wherein the content rights comprise at least one economic right for an accounting for the content stream, further comprising the step of accounting for the content stream based on the content rights.

119. The method according to claim 118, wherein a content owner receives consideration for content.

120. The method according to claim 118, wherein a viewer receives consideration for content.

121. The method according to claim 118, wherein a publisher receives consideration for content.

122. The method according to claim 118, wherein said accounting is performed on-line.

123. The method according to claim 122, wherein said accounting requires handshake verification.

124. The method according to claim 118, wherein said accounting is performed in a batch mode.

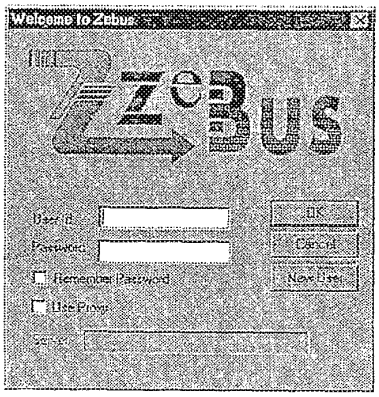

Welcome to Zebus
ZEBUS
User ID
Password
OK
Cancel
New User
Remember Password
Use Proxy

Fig. 1

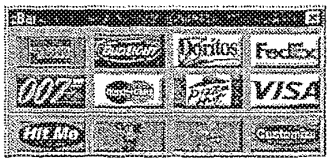


Fig 2

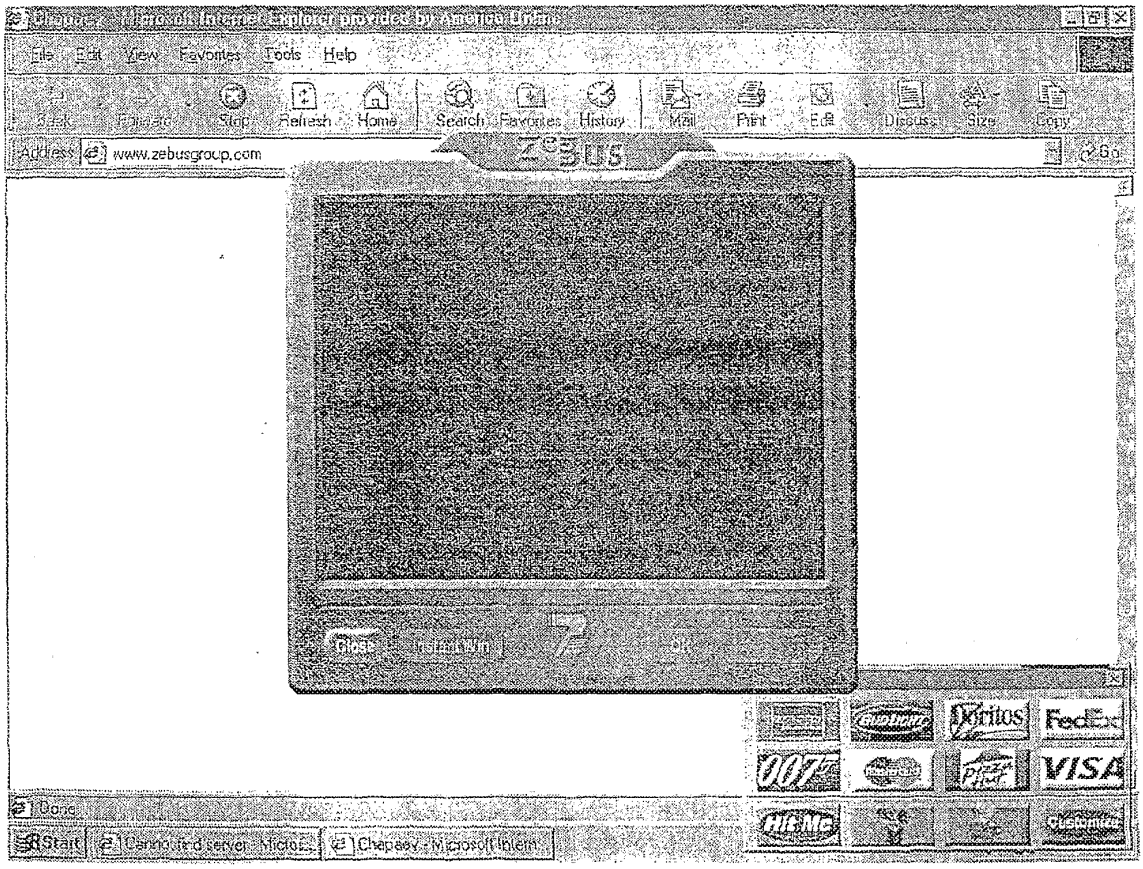

File Edit View Favorites Tools Help
Stop Refresh Home Search Favorites History Mail Print Edit Discuss Size Copy
Address www.zebusgroup.com
ZEBUS
Close
Doritos
FedEx
VISA
Start

Fig. 3

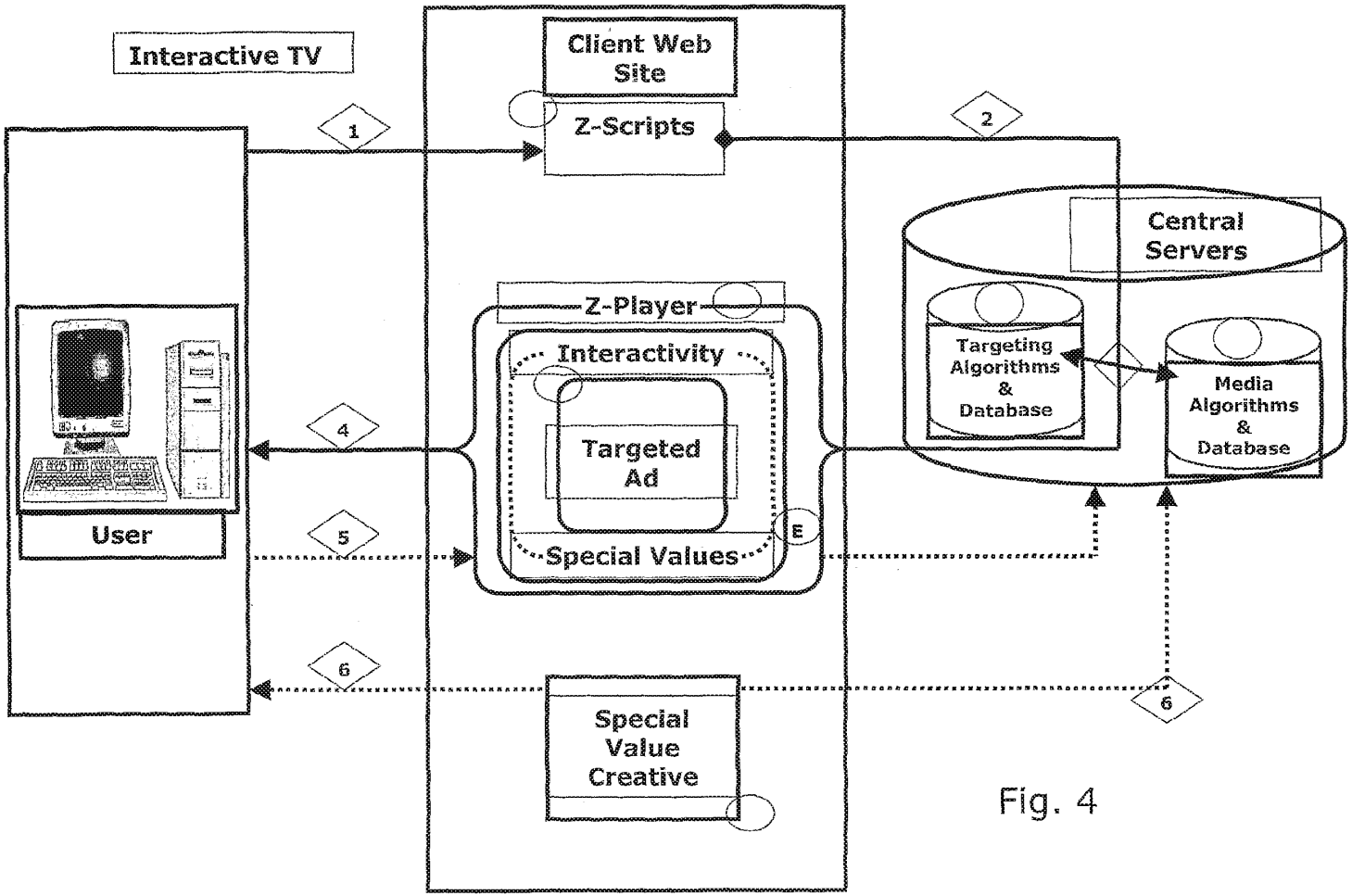

Interactive TV
Client Web Site
Z-Scripts
1
2
Central Servers
Z-Player
Interactivity
Targeting Algorithms & Database
Media Algorithms & Database
Targeted Ad
4
User
5
Special Values
E
6
6
Special Value Creative

Fig. 4

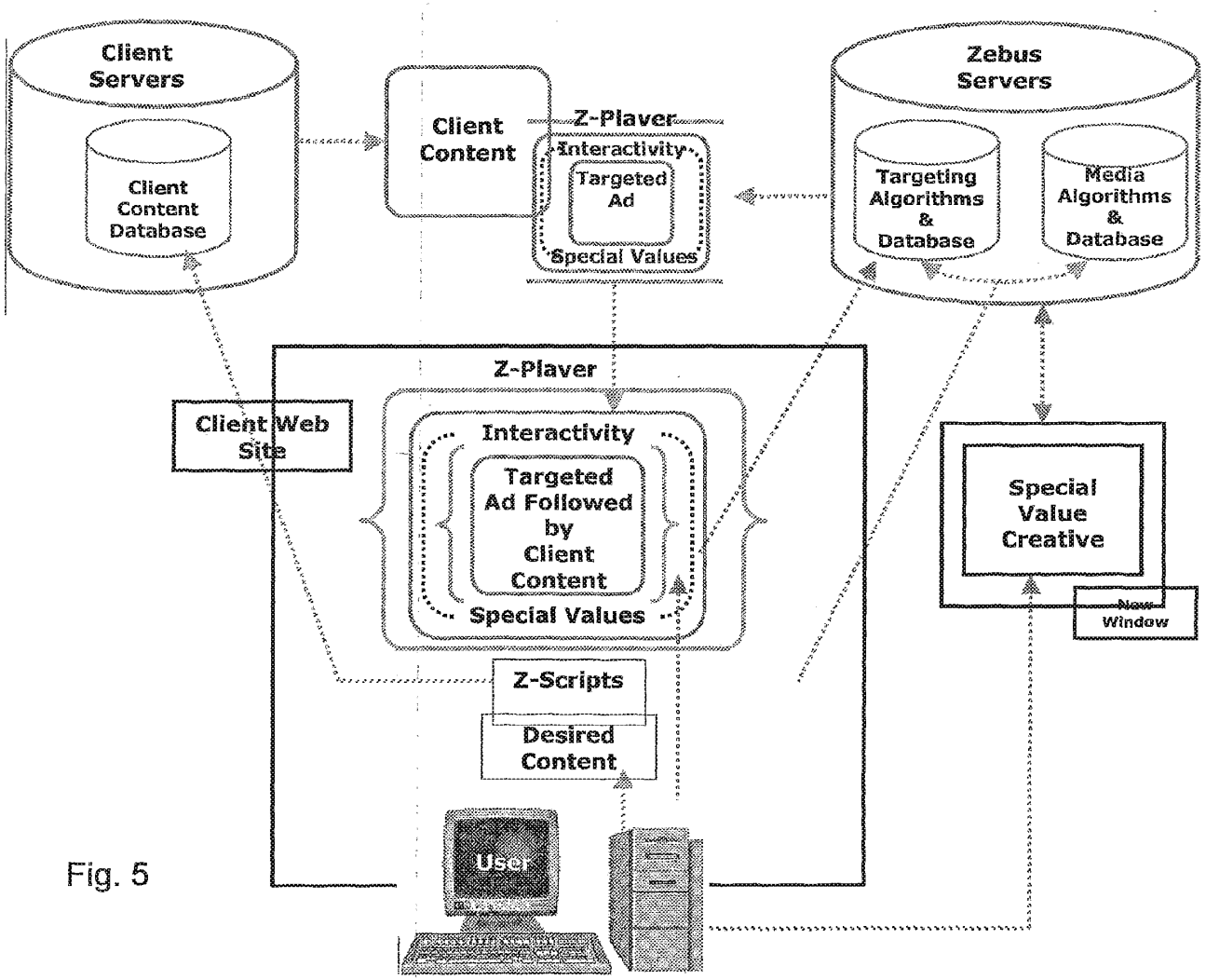

Client Servers
Client Content Database
Client Content
Z-Player
Interactivity
Targeted Ad
Special Values
Zebus Servers
Targeting Algorithms & Database
Media Algorithms & Database
Z-Player
Client Web Site
Interactivity
Targeted Ad Followed by Client Content
Special Values
Special Value Creative
New Window
Z-Scripts
Desired Content
User

Fig. 5

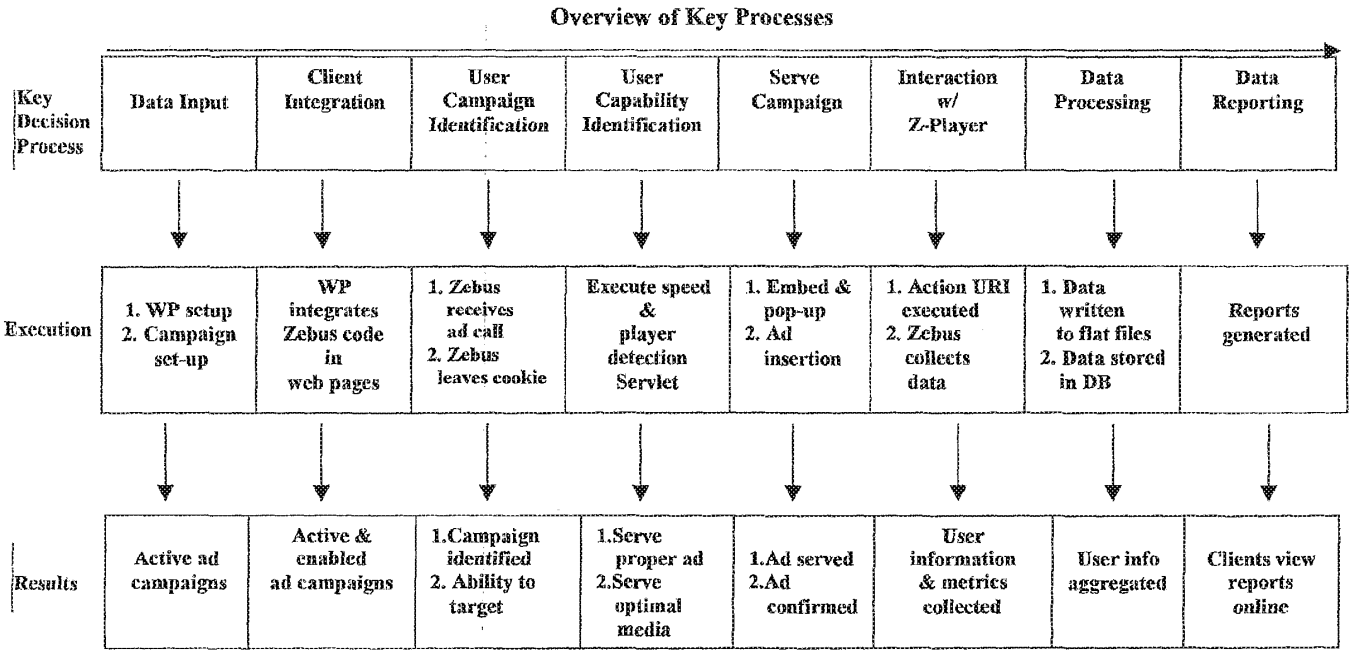
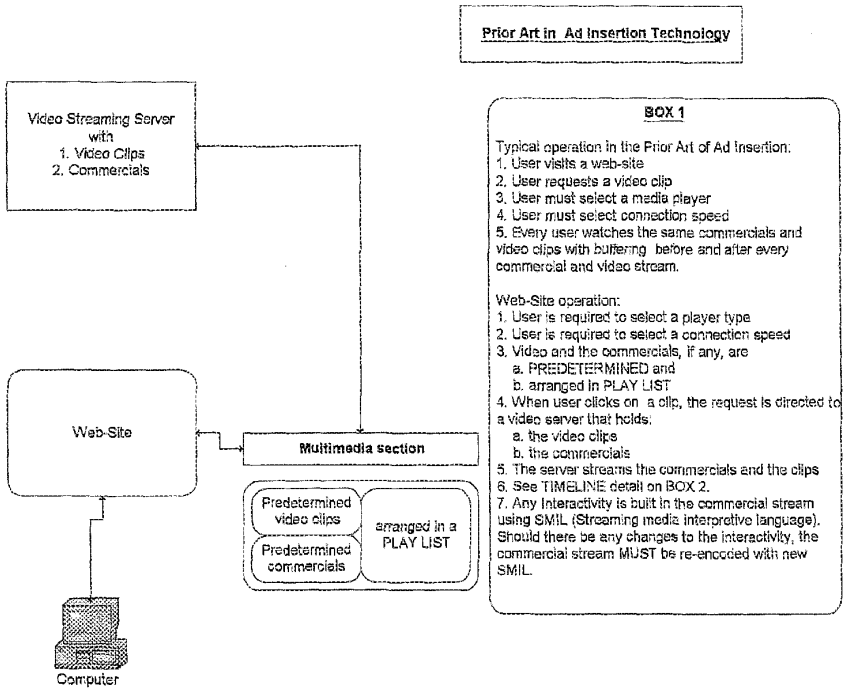


Fig. 6

Google Exhibit No. 1008
Page 336



BOX 2

Assume that a website will stream the following streaming media to a user in one con-current session:

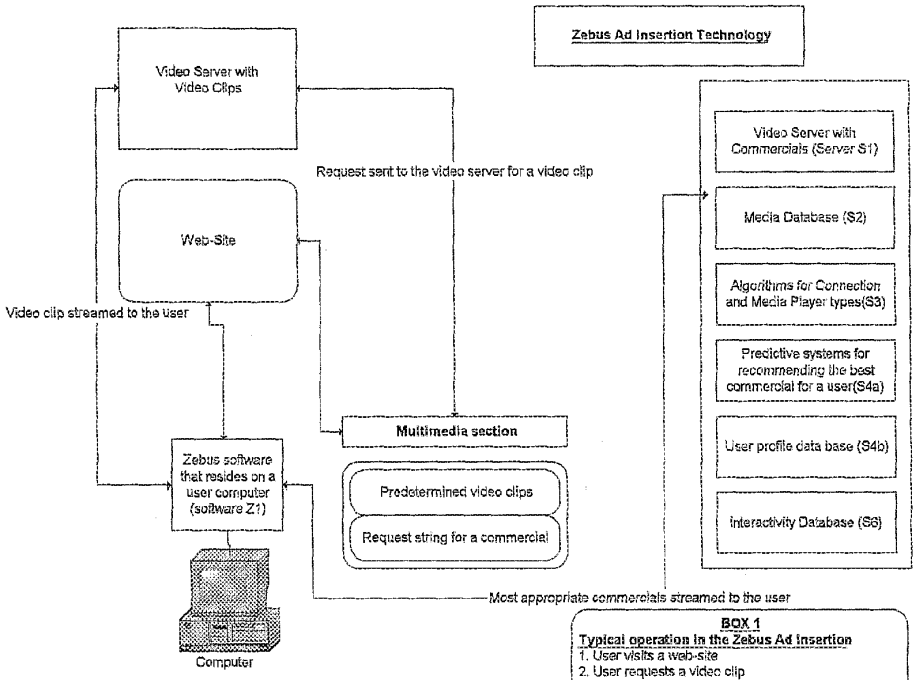
1. Commercial C1 with duration T1
2. Video V1 with duration T2
3. Commercial C2 with Duration T3
4. Video V2 with Duration T4

Since every clip must be buffered before it can be played, assume that the AVERAGE buffer time is B1. It takes an average of B2 for a stream to begin buffering. The above 4 video streams together form the VIDEO.

User requests the VIDEO at time t=0.
C1 starts buffering at t = B2
C1 finishes buffering at t= B2 + B1
C1 begins playing at t=B2+B1
C1 ends playing at t=B1+B2+T1
V1 begins buffering at t= B1+ B2 + T1+ B2
V1 ends buffering at t= B1+ 2B2 + T1+ B1
V1 starts playing at t= 2B1+2B2 +T1
V1 ends playing at t = 2B1 + 2B2 +T1+ T2

C2 starts buffering at t = 2B1 + 2B2 +T1+ T2 + B2
C2 finishes buffering at t=2B1 + 3B2 +T1+ T2 + B1
C2 begins playing at t= 3B1+3B2+T1+T2
C2 ends playing at t = 3B1+3B2+T1+T2 + T3
V2 begins buffering at t = 3B1+3B2+T1+T2 + T3 + B2
V2 ends buffering at t= 3B1+4B2+T1+T2 + T3 + B1
V2 starts playing at t= 4B1+4B2+T1+T2 + T3
V2 ends playing at t = 4B1 + 4B2 + T1 + T2 + T3 + T4

Fig. 7
PRIOR ART



BOX 2

Assume that a website will stream the following streaming media to a user in one concurrent session:

1. Commercial C1 with duration T1
2. Video V1 with duration T2
3. Commercial C2 with Duration T3
4. Video V2 with Duration T4

Since every clip must be buffered before it can be played, assume that the AVERAGE buffer time is B1. It takes an average of B2 for a stream to begin buffering. The above 4 video streams together form the VIDEO.

Zebus Process

User requests the VIDEO at time t=0.
C1 starts buffering at t = B2
C1 finishes buffering at t= B2 + B1
C1 begins playing at t=B2+B1
C1 ends playing at t=B1+B2+T1
V1 begins buffering at t= B1+ B2 + T1 - B2- B1
V1 ends buffering at t= T1+B1+B2
V1 starts playing at t= T1+B1+B2
V1 ends playing at t = T1+ T2+B1+B2

C2 starts buffering at t = T1+ T2+B1+B2- B2 - B1
C2 finishes buffering at t=T1+ T2 +B1+B2
C2 begins playing at t= T1+T2+B1+B2
C2 ends playing at t = T1+T2 + T3+B1+B2
V2 begins buffering at t = T1+T2 + T3 +B1+ B2- B2 - B1
V2 ends buffering at t= T1+T2 + T3+B1+B2
V2 starts playing at t= T1+T2 + T3 +B1+B2
V2 ends playing at t = T1 + T2 + T3 + T4+B1+B2

BOX 1

Typical operation in the Zebus Ad insertion

1. User visits a web-site
2. User requests a video clip

Web-Site operation:
1. Video is predetermined
2. A request for commercial clip is made (not preset)

Zebus Technology
1. Zebus detects the player type (S3)
2. Zebus detects the connection speed (S3)
3. Zebus decides the best format for the commercials (S2)
4. Zebus decides the timing of the commercials as to when the commercials will be shown. (S4)
5. Zebus decides the best commercial for the individual (S4a and S4b)
6. Zebus decides on the interactivity (S6)
7 Zebus sends that information to the Zebus software residing on a user's machine (Z1)

Zebus software (Z1)

When the user clicks on a clip, the request is directed to a video server that holds:
a. the video clips
5. The Zebus software call the video stream and the commercial clips
6. Starts the processes according to the information received in Zebus technology (4,5)
7. Zebus plays the commercials and the stream according to the timeline . See TIMELINE detail in box 2.
7. Any interactivity is built AROUND the commercial stream using DATABASE . Should there be any changes to the interactivity, simply change the database table

Fig. 8

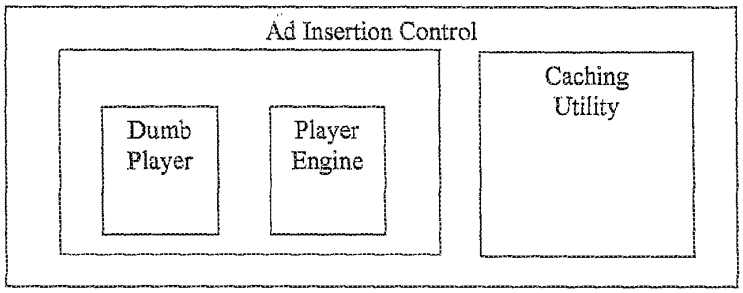

Ad Insertion Control
Dumb Player
Player Engine
Caching Utility

Fig. 9

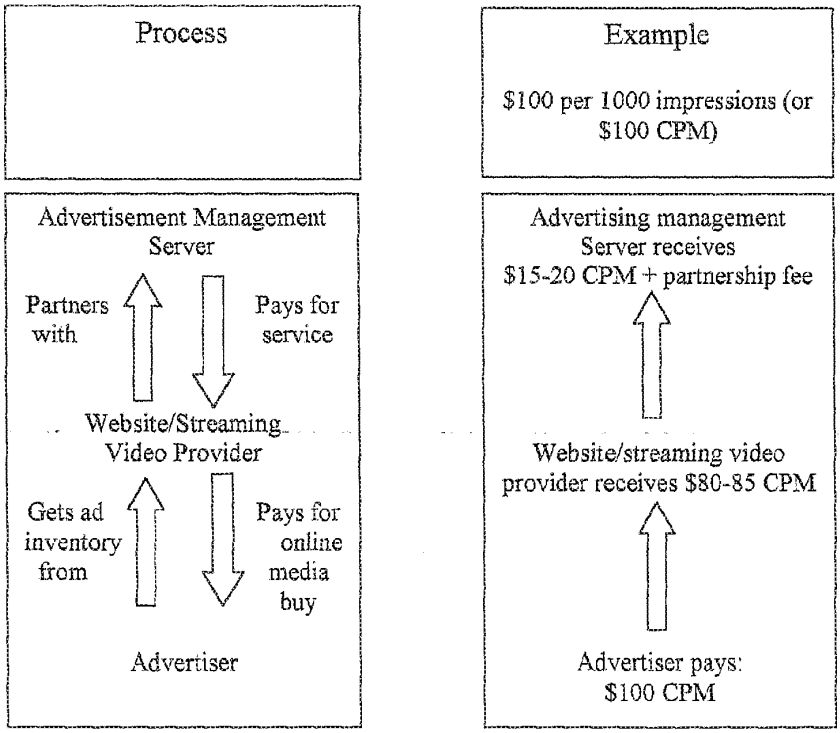

Process
Example
$100 per 1000 impressions (or $100 CPM)
Advertisement Management Server
Partners with
Pays for service
Website/Streaming Video Provider
Gets ad inventory from
Pays for online media buy
Advertiser
Advertising management Server receives $15-20 CPM + partnership fee
Website/streaming video provider receives $80-85 CPM
Advertiser pays: $100 CPM

Fig. 10

Pop-up Player with Wallpaper

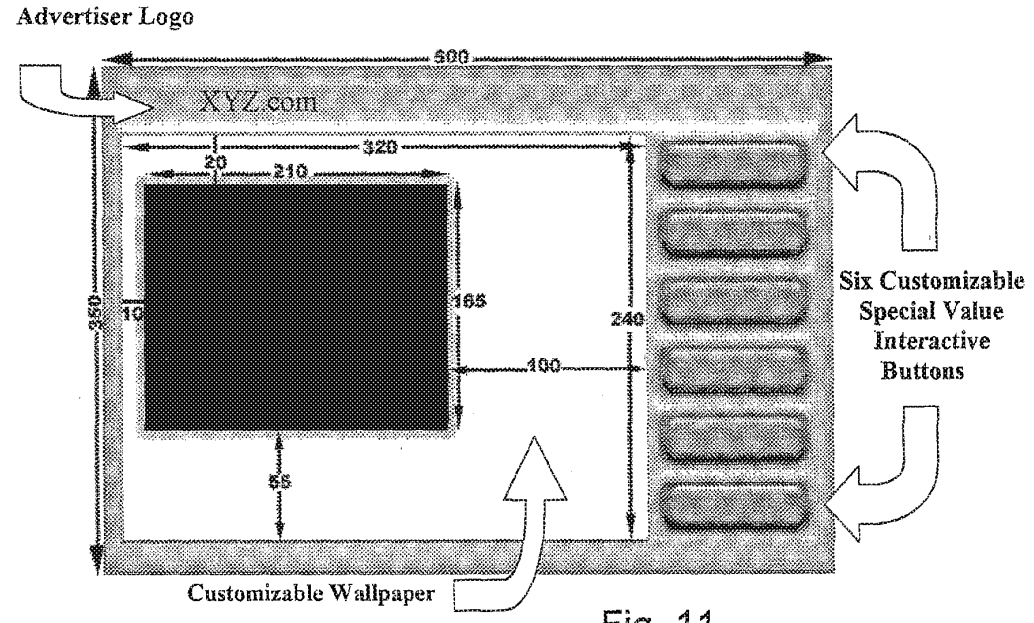


Fig. 11

Pop-up Player without Wallpaper

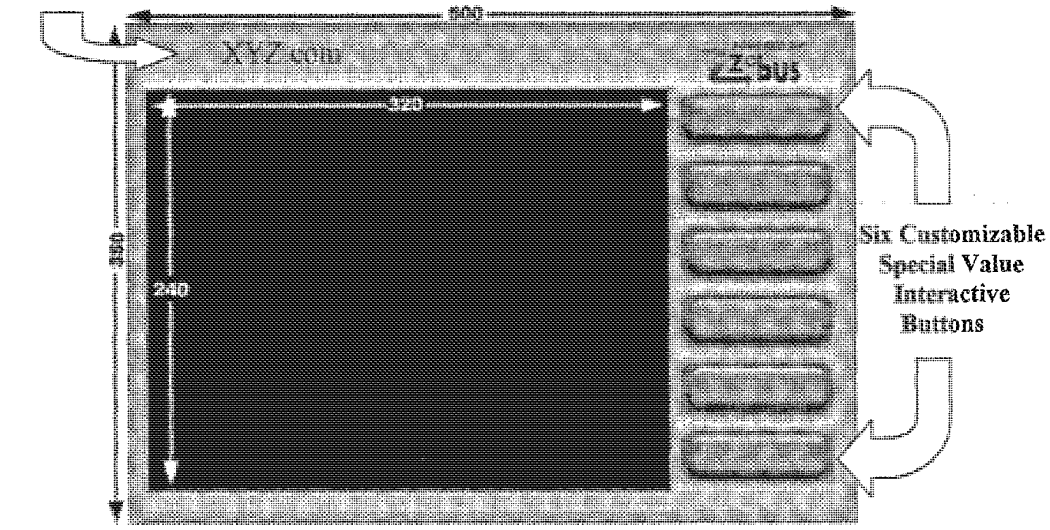


Fig. 12

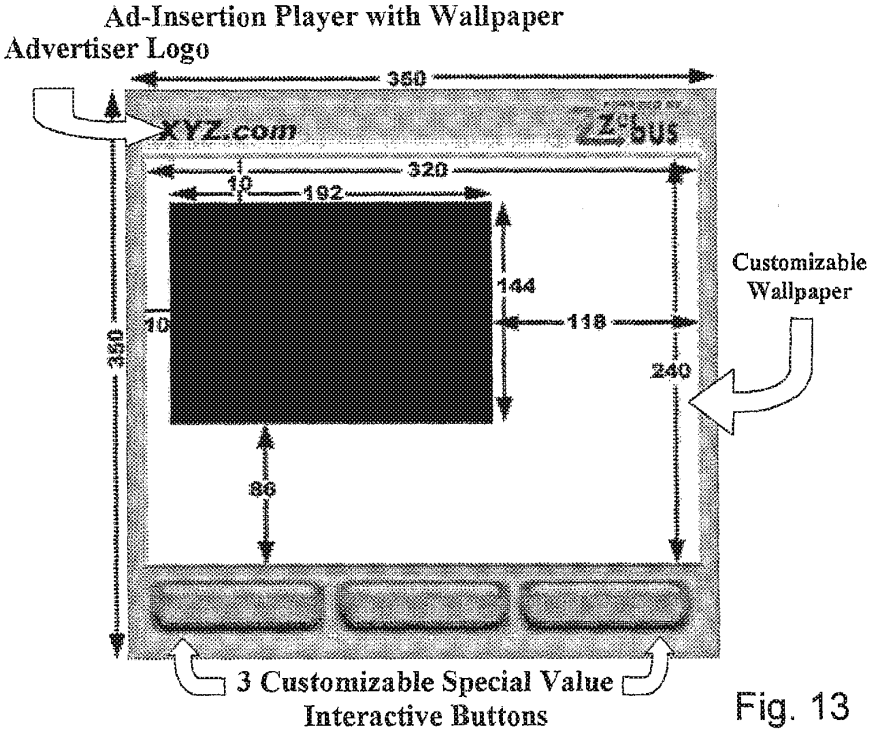

Ad-Insertion Player with Wallpaper
Advertiser Logo
350
XYZ.com
320
10
192
144
118
240
86
350
10
Customizable
Wallpaper
3 Customizable Special Value
Interactive Buttons

Fig. 13

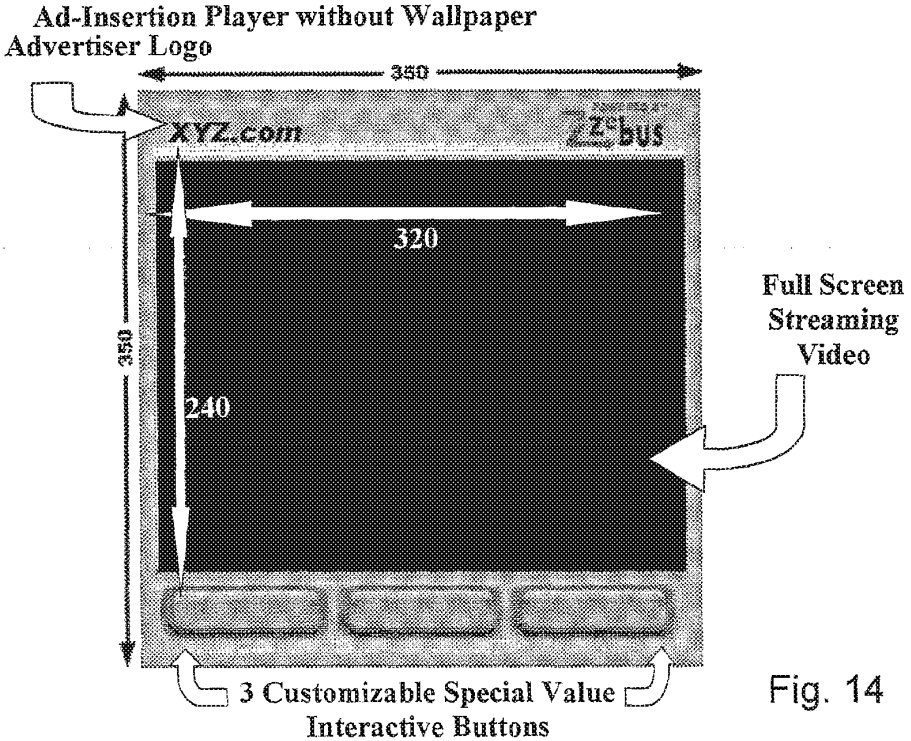

Ad-Insertion Player without Wallpaper
Advertiser Logo
350
XYZ.com
320
350
240
Full Screen
Streaming
Video
3 Customizable Special Value
Interactive Buttons

Fig. 14

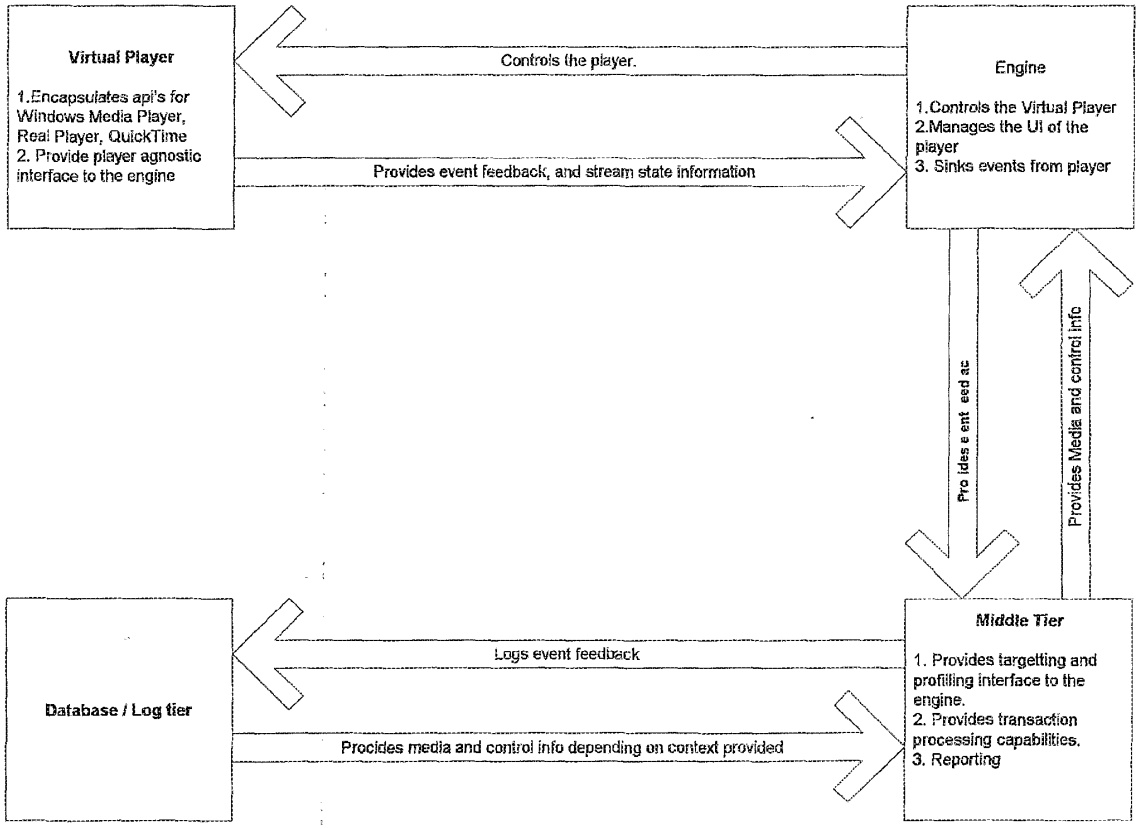

Virtual Player
1.Encapsulates api's for Windows Media Player, Real Player, QuickTime
2. Provide player agnostic interface to the engine
Controls the player.
Provides event feedback, and stream state information
Engine
1.Controls the Virtual Player
2.Manages the UI of the player
3. Sinks events from player
Pro ides e ent eed ac
Provides Media and control info
Database / Log tier
Logs event feedback
Procides media and control info depending on context provided
Middle Tier
1. Provides targetting and profilling interface to the engine.
2. Provides transaction processing capabilities.
3. Reporting

Fig. 15

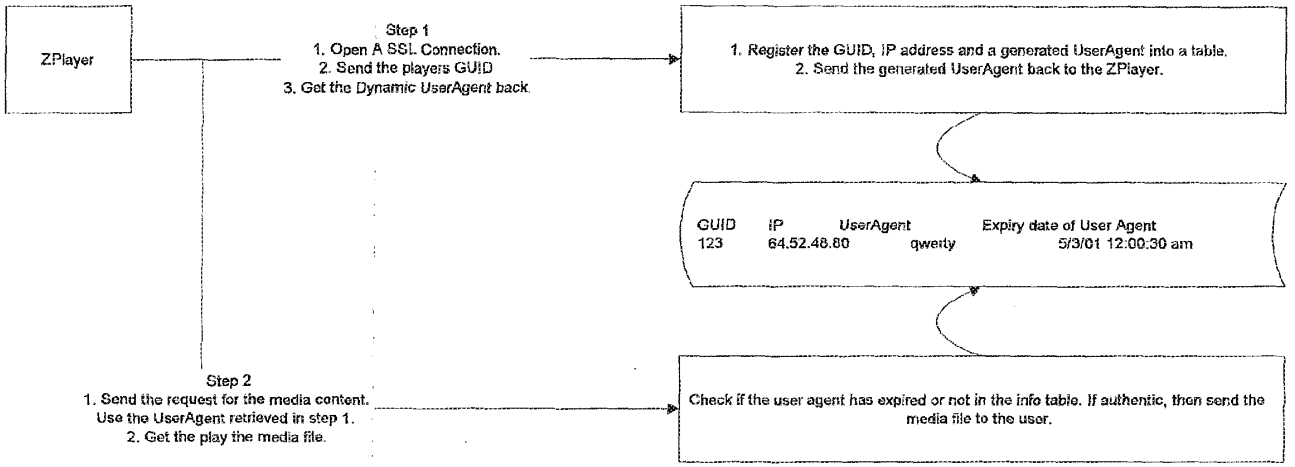

ZPlayer
Step 1
1. Open A SSL Connection.
2. Send the players GUID
3. Get the Dynamic UserAgent back.
1. Register the GUID, IP address and a generated UserAgent into a table.
2. Send the generated UserAgent back to the ZPlayer.
GUID IP UserAgent Expiry date of User Agent
123 64.52.48.80 qwerty 5/3/01 12:00:30 am
Step 2
1. Send the request for the media content.
Use the UserAgent retrieved in step 1.
2. Get the play the media file.
Check if the user agent has expired or not in the info table. If authentic, then send the media file to the user.

Fig. 16

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/17259

A. CLASSIFICATION OF SUBJECT MATTER
IPC(7) : G06F 17/60
US CL : 705/14
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. : 705/14

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)
EAST search of USPAT, US-PGPUB, EPO, JPO, Derwent WPI and IBM TDB databases.

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 5,970,473 A (GERSZBERG et al.) 19 October 1999 (19.10.1999), col. 8 lines 22-29.	1-21 --- 112
X	US 6,006,257 A (SLEZAK) 21 December 1999 (21.12.1999), col. 4 lines 2-4.	22 and 111
X	US 5,732,216 A (LOGAN et al.) 24 March 1998 (24.03.1998), col. 17 lines 49-51, col. 4 line 48, col. 3 lines 18-25, col. 9 lines 23-25 and col. 27 lines 27-29.	23-93 and 96-110
X,P	US 6,195,680 B1 (GOLDSZMIDT et al.) 27 February 2001 (27.02.2001), col. 14 lines 1-19.	94, 95
Y	US 5,991,306 A (BURNS et al.) 23 November 1999 (23.11.1999), col. 4 lines 57-60.	112
A	US 4,890,319 A (SETH-SMITH et al.) 26 December 1989 (26.12.1989).	113-124

☐ 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
18 September 2001 (18.09.2001)	16 NOV 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 Eric Stamber Telephone No. 703-305-3900

Form PCT/ISA/210 (second sheet) (July 1998)

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 411 698 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
21.04.2004 Bulletin 2004/17

(51) Int Cl.[7]: **H04L 29/06**

(21) Application number: **03027277.7**

(22) Date of filing: **21.06.2001**

(84) Designated Contracting States:
DE FR GB

(30) Priority: **22.06.2000 US 603108**

(62) Document number(s) of the earlier application(s) in accordance with Art. 76 EPC:
01948643.0 / 1 297 676

(71) Applicant: **APPLE COMPUTER, INC.**
Cupertino, CA 95014 (US)

(72) Inventors:
- **Serenyi, Denis**
  Menlo Park, CA 94025 (US)
- **Lecroy, Chris**
  Boulder Creek, CA 95006 (US)

(74) Representative:
Jehle, Volker Armin, Dipl.-Ing. et al
Patentanwälte
Bosch, Graf von Stosch, Jehle,
Flüggenstrasse 13
80639 München (DE)

Remarks:
This application was filed on 28 - 11 - 2003 as a divisional application to the application mentioned under INID code 62.

(54) Methods and apparatuses for transferring data

(57) The present invention provides several methods and apparatuses for transmitting multimedia data using streaming media protocols such as real-time transfer protocols (RTP) and real-time streaming protocols (RTSP) in a computer network environment. In one exemplary embodiment, a request for RTP data and its associated extension is sent from the caching proxy server (401) to the server (301). The request may be for one specific type of data or multiple unrelated types of data. The server (301) responds to the request indicating its support for the requested RTP extension data. The caching proxy server (401) determines whether to proceed or terminate the data transmission process based on the response provided by the server (301). If it is determined to proceed with the data transmission process, the caching proxy (401) informs the server (301) to send the requested and supported RTP data. The server (301) sends the requested data in a variable and extendible header format.

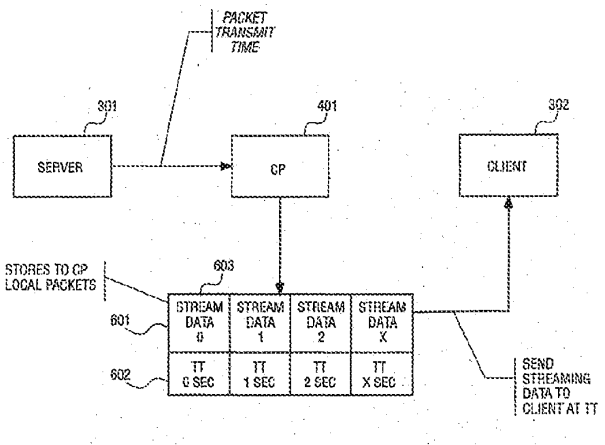


FIG. 6

Printed by Jouve, 75001 PARIS (FR)

EP 1 411 698 A1

Description

FIELD OF THE INVENTION

[0001] The present invention relates to the field of multimedia data transmission. In particular, the present invention in one exemplary embodiment relates to multimedia data transmission of real-time transfer protocol (RTP) packets using real time streaming protocol (RTSP) in a computer network environment.

INTRODUCTION AND BACKGROUND OF THE INVENTION

[0002] Methods of transmitting data are commonly known and performed today on a routine basis to send various multimedia data such as text, graphics, audio, video, images etc. across computer networks situated in various parts of the world. Generally the transmission process requires both hardware and software for performing its function. Typically, the hardware includes various types of personal computers and hand held multimedia data sending or receiving devices. These devices run under the control of an operating system and utilize multimedia application software programs. As is known in the art, streaming media data is data which is transmitted to a receiving computer system and presented (usually after buffering temporarily at the receiving system) and then discarded (not stored) at the receiving system.

[0003] Currently, data is sent in form of packets from one multimedia device to another. A large amount of information is required to be sent in a real-time manner in the data packets, which imposes a heavy load on the systems. Streaming media data, such as Real-Audio data in streaming media format specified by Real-Networks, is sent through the Internet is near real-time manner in many cases.

[0004] In one approach, the components involved in data transmission of streaming media are known to be a server (which may be referred to as originating server), a caching proxy server and a client. These components in various combinations communicate with each other for transmitting data packets in real-time. The communication link that currently exists between the components uses real-time transfer protocols (RTP) and real-time streaming protocols (RTSP) to communicate and send packets to each other. For this approach to work, a caching proxy server needs to communicate with the system server, receive a stream of RTP data packets, and transfer the information contained within the RTP data packets to a client. Figure 1a shows an example of a prior method in which a caching proxy server receives streaming media data and provides this data to a client. In order to perform its function properly and efficiently, the caching proxy server needs several pieces of information from the server to be able to cache an RTP stream easily and reliably.

[0005] A problem with the current approach is that it is not able to provide some of the key required information such as data packet transmit time and video packet frame type information that a caching proxy needs to be efficient. This information allows a caching proxy server to provide smooth packet delivery to its client by knowing the time an RTP data packet was intended to be sent, and type of video frame that is being sent without knowing the specific payload format. Another problem with the current approach is that it is not able to provide multiple pieces of unrelated data in one delivery to the caching proxy server. Furthermore, packets from the server may be "lost" and never reach the caching proxy server. In addition, there is normally no way to recreate a complete "pristine" copy at the caching proxy server.

[0006] Prior art servers communicate RTP information to the caching proxy server by sending information through a cache-control header. In one approach, a cache-control header contains normal header fields. In another approach, unrelated to cache control of RTP information, a single type of additional information has been added to the normal fields in a header extension format without specifying the type of additional information. In this approach only a single piece of RTP extension can be added to the normal field of the header and sent at any one time.

[0007] A problem with using this limited, non-extensible approach is that a server is not able to attach multiple sets of unrelated data at a time to send to the caching proxy server. Another problem with this approach is that the header extension used in these methods are still not able to provide all the information a caching proxy server needs to cache a stream properly and to transmit the stream properly. Yet another problem with this approach is that there is no way to identify the particular extension independently of other possible extensions.

SUMMARY OF THE INVENTION

[0008] The present invention provides several methods and apparatuses for transmitting multimedia data using streaming media protocols such as real-time transfer protocols (RTP) and real-time streaming protocols (RTSP) in a computer network environment. In one exemplary embodiment, a request for RTP data is sent from the caching proxy server to the server. The request may be for one specific type of data and its related extensions or multiple unrelated types of data and their related extensions. The server responds to the request indicating its support for the requested RTP data. The caching proxy server determines whether to proceed or terminate the data transmission process based on the response provided by the server. If it is determined to proceed with the data transmission process, the caching proxy informs the server to send the requested and supported RTP data. The server sends the requested data in a variable and extendible header format.

[0009] In another embodiment, the caching proxy

Google Exhibit No. 1008
Page 346

server requests and receives packet transmit time data and / or packet frame type data from the server. The caching proxy server uses the frame type data to communicate with the client and supply frames based on client's capacity to handle loads at given times. Transmit time data is also used by the caching proxy to store packets locally and deliver these packets at appropriate times to the client for a smooth packet delivery.

[0010] According to another aspect of the invention, a method for operating a caching proxy server comprises:

sending a request for streaming media data to a server, said request including a request for data associated with said streaming media data, said request including an identifier which represents one of several possible types of data associated with said streaming media data;
receiving a response from the server indicating support for the requested streaming media data;
informing the server to send the supported data associated with said streaming media data;
receiving the streaming media data from the server;
receiving a request from the client to send streaming media data; and
sending the requested streaming media data to the client.

[0011] Advantageously, said receiving and sending uses a real-time transport protocol (RTP).

[0012] Advantageously, said receiving streaming media data from the server is in an extensible extended header format.

[0013] Advantageously, said sending a request may be for one or more various and unrelated types of streaming media data to be sent at a time.

[0014] Advantageously, said response from the server comprising response for each supported type of streaming media and no response for any unsupported types of streaming media data.

[0015] Advantageously, said extensible extended header format is appended before sending to client.

[0016] Advantageously said appending comprises stripping of name and ID part of the extensible extended header.

[0017] Advantageously the method further comprises determining if a requested type of streaming media data, that which is required by a caching proxy server to be able to perform its processes, is missing in the response by the server.

[0018] Advantageously the method further comprises terminating the data transmission process if the requested type of streaming media data is missing in server's response and is critical to the data transmission process.

[0019] According to another aspect of the invention, a method for extending an RTP header comprises:

adding a first RTP sub-extension ID to an RTP header;
defining a length of said first RTP sub-extension by providing a sub-extension length;
providing data corresponding to the RTP sub-extension ID within said length defined for said first RTP sub-extension; and
having subsequent RTP sub-extensions following the first RTP sub-extension.

[0020] Advantageously the length of the RTP sub-extension being defined by a whole number of 32 bit words.

[0021] Advantageously the first RTP sub-extension immediately following the RTP header.

[0022] Advantageously the RTP sub-extension length is immediately followed by RTP sub-extension data and immediately preceded by RTP sub-extension ID.

[0023] Advantageously the RTP sub-extension contains transmit time information of each RTP packet.

[0024] Advantageously the RTP sub-extension contains persistent ID information.

[0025] Advantageously the RTP sub-extension contains Frame Type information.

[0026] Advantageously the frame type being a 16-bit unsigned integer value representing a different frame for each value.

[0027] Advantageously unsigned integer and frame type comprises:

assigning an integer value of "0" to an unknown frame type; an integer value of "1" to a key frame type; an integer value of "2" to a p-frame type; and an integer value of "3" to a b-frame type.

[0028] Advantageously said key-frame being most important in priority than any other frames.

[0029] Advantageously said p-frame being less important in priority than key-frame, and more important in priority than b-frame.

[0030] Advantageously said b-frame being less important in priority than p-frame.

[0031] Advantageously said b-frame being less important in priority than key-frame.

[0032] Advantageously said unknown-frame being either more or less important in priority than key-frame, p-frame and b-frame.

[0033] Advantageously said key-frame being more important in priority than p-frames, b-frames, and any other frames.

[0034] According to another aspect of the invention a method of negotiating for various types of streaming media data by the server comprises:

receiving a request for one or more types of streaming media data from a caching proxy server or a client, said request including a request for data associated with said streaming media data, said request

Google Exhibit No. 1008
Page 347

including an identifier which represents one of several possible types of data associated with said streaming media data;
determining if requested types of streaming media data are supported by the server; and
responding to the request with a response indicating the capability of the server to support the request.

[0035] Advantageously the method further comprises receiving a request to send supported RTP extensions to the caching proxy or the client.

[0036] Advantageously the method further comprises responding to request to send and sending all the supported and requested extensions.

[0037] Advantageously the method further comprises receiving a command terminating the negotiation process.

[0038] According to another aspect of the invention a method of negotiating for various types of streaming media data by the caching proxy server comprises:

sending a request for one or more types of related or unrelated streaming media data to a server, said request including a request for data associated with said streaming media data, said request including an identifier which represents one of several possible types of data associated with said streaming media data;
receiving a response to each requested type of streaming media data; and
deciding whether to proceed or terminate negotiation process associated with streaming media data.

[0039] Advantageously the deciding comprises:

determining if any requested type of streaming media data is not supported by server;
checking if unsupported type of streaming media data is essential to caching proxy server operations; and
sending an execution command to server.

[0040] Advantageously determining supported types of streaming media data being performed by checking if a response in a form of an echo or otherwise has been sent for requested type of streaming media data.

[0041] Advantageously the execution command being sent based on results of checking if unsupported type of streaming media data is essential to caching proxy server operations.

[0042] Advantageously the decision being to terminate negotiation process.

[0043] Advantageously the decision being to proceed with negotiation process and requesting the server to send remaining supported type of streaming media data.

[0044] According to another aspect of the invention a method of frame thinning by caching proxy server comprises:

receiving a message from a client, said message indicating a need to thin streaming media data being sent to said client;
evaluating priority of streaming media data; and
sending only selected streaming media data.

[0045] Advantageously said evaluating comprises:

naming unsigned integers to frame types;
assigning an integer value of "0" to an unknown frame type; an integer value of "1" to a key frame type; an integer value of "2" to a p-frame type; and an integer value of "3" to a b-frame type.

[0046] Advantageously said key-frame being most important in priority than any other frames.

[0047] Advantageously said p-frame being less important in priority than key-frame, and more important in priority than b-frame.

[0048] Advantageously said b-frame being less important in priority than p-frame.

[0049] Advantageously said b-frame being less important in priority than key-frame.

[0050] Advantageously said unknown-frame being either more or less important in priority than key-frame, p-frame and b-frame.

[0051] Advantageously said key-frame being more important in priority than p-frames, b-frames, and any other frames.

[0052] Advantageously the method further comprises:

receiving a second message from a client to further thin streaming media data;
processing message and eliminating more selected streaming media data and sending streaming media data of higher priority.

[0053] Advantageously said selected streaming media frame being eliminated being a b-frame, and sending streaming media data with higher priority than b-frame to the client.

[0054] Advantageously said streaming media data being eliminated being a p-frames and b-frame, and sending frames with higher priority than both p-frames and b-frames to the client.

[0055] According to another aspect of the invention a method of frame thinning by client comprises:

sending a message to a caching proxy server, said message indicating a need to thin streaming media data received at said client;
receiving media back from said caching proxy server that are higher in order than low order streaming media data.

Google Exhibit No. 1008
Page 348

[0056] Advantageously the method further comprises:

sending a subsequent message from a client to further thin streaming media data;
receiving streaming media data that is higher in order then previous streaming media data received.

[0057] Advantageously the method further comprises:

assigning an unsigned integer to a frame associated with streaming media data, wherein said assigning further comprising assigning an integer value of "0" to an unknown frame type; an integer value of "1" to a key frame type; an integer value of "2" to a p-frame type; and an integer value of "3" to a b-frame type.

[0058] Advantageously said key-frame being most important in priority than any other frames.

[0059] Advantageously said p-frame being less important in priority than key-frame, and more important in priority than b-frame.

[0060] Advantageously said b-frame being less important in priority than p-frame.

[0061] Advantageously said b-frame being less important in priority than key-frame.

[0062] Advantageously said unknown-frame being either more or less important in priority than key-frame, p-frame and b-frame.

[0063] Advantageously said key-frame being more important in priority than p-frames, b-frames, and any other frames.

[0064] Advantageously said sending comprises eliminating p-frames and sending selected streaming media data that is higher in order than p-frames to the client.

[0065] Advantageously said sending comprises eliminating both p-frames and b-frames, and sending selected streaming media data that is higher in order than both p-frames and b-frames.

[0066] According to another aspect of the invention a machine-readable medium that provides executable instructions, which when executed by a set of processors, cause said set of processors to perform operations for producing a streaming media data at a caching proxy server comprising:

transmitting a request from for streaming media data to be delivered to said caching proxy server;
transmitting a request for data associated with said streaming media data, said request including an identifier which represents one of several possible types of data associated with said streaming media data;
receiving said streaming media data and storing said streaming media data on a storage device which is capable of being controlled by said caching proxy server; and
receiving said data associated with said streaming media data.

[0067] Advantageously the machine-readable medium further comprises:

storing said data associated with said streaming media data in said storage device.

[0068] According to another aspect of the invention a machine-readable medium that provides executable instructions, which when executed by a set of processors, cause said set of processors to perform data transmission operations from a server data processing system comprising:

receiving a request for streaming media data, said request including a request for data associated with said streaming media data, said request including an identifier which represents one of several possible types of data associated with said streaming media data;
responding to the request with a response indicating a capability of said server to support the request; and
sending the requested data associated with said streaming media data.

[0069] Advantageously said sending uses a real-time transport protocol (RTP).

[0070] Advantageously said request may be made by a caching proxy server or a client.

[0071] Advantageously said responding with a response occurring only if said server supports the request.

[0072] Advantageously the machine-readable medium further comprises sending the requested data associated with the transmission protocol in an extensible extended header format.

[0073] Advantageously said extensible extended header comprises an extension name and an extension identification (ID) associated with each separate RTP extension.

[0074] Advantageously said request may be for one or more type of transmission protocol data at a time.

[0075] Advantageously said response by the server comprising response for each supported transmission protocol data and no response for any unsupported transmission protocol data.

[0076] Advantageously said machine-readable medium further comprises receiving a request to send the transmission protocol data after sending a response for supported data, and sending only the requested and supported transmission protocol data.

[0077] According to another aspect of the invention a machine-readable medium that provides executable instructions, which when executed by a set of processors,

Google Exhibit No. 1008
Page 349

cause said set of processors to perform data transmission receiving operations from a sever comprising:

sending a request for streaming media data to said server, said request including a request for data associated with said streaming media data, said request including an identifier which represents one of several possible types of data associated with said streaming media data;
receiving a response from said server indicating support for the requested streaming media data;
informing said server to send the supported data associated with said streaming media data;
receiving the supported streaming media data from said server;
receiving a request from a client to send streaming media data; and
sending the requested streaming media data to said client.

[0078] Advantageously said receiving and sending uses a real-time transport protocol (RTP).

[0079] Advantageously said receiving streaming media data from the server is in an extensible extended header format.

[0080] Advantageously said sending a request may be for one or more various and unrelated types of streaming media data to be sent at a time.

[0081] Advantageously said response from the server comprising response for each supported type of streaming media and no response for any unsupported types of streaming media data.

[0082] Advantageously said extensible extended header format is appended before sending to client.

[0083] Advantageously the appending comprises stripping of name and ID part of the extensible extended header.

[0084] Advantageouslysaid machine-readable medium further comprises determining if a requested type of streaming media data, that which is required by a caching proxy server to be able to perform its processes, is missing in the response by the server.

[0085] Advantageously said machine-readable medium further comprises terminating the data transmission process if the requested type of streaming media data is missing in server's response and is critical to the data transmission process.

[0086] According to another aspect of the invention a machine-readable medium that provides executable instructions, which when executed by a set of processors, cause said set of processors to perform RTP header extending operations comprises:

adding a first RTP sub-extension ID to an RTP header;
defining a length of the first RTP sub-extension by providing a sub-extension length;
providing data corresponding to the RTP sub-extension ID within said length defined for said first RTP sub-extension; and
having subsequent RTP sub-extensions following the first RTP sub-extension.

[0087] Advantageously said length of said RTP sub-extension being defined by a whole number of 32 bit words.

[0088] Advantageously said first RTP sub-extension immediately follows the RTP header.

[0089] Advantageously said RTP sub-extension length is immediately followed by RTP sub-extension data and immediately preceded by RTP sub-extension ID.

[0090] Advantageously said RTP sub-extension contains transmit time information of each RTP packet.

[0091] Advantageously said RTP sub-extension contains persistent ID information.

[0092] Advantageously said RTP sub-extension contains RTP Frame Type information.

[0093] Advantageously said frame type being a 16-bit unsigned integer value representing a different frame for each value.

[0094] Advantageously said unsigned integer and frame type further comprising steps of:

assigning an integer value of "0" to an unknown frame type; an integer value of "1" to a key frame type; an integer value of "2" to a p-frame type; and an integer value of "3" to a b-frame type.

[0095] Advantageously said key-frame being most important in priority than any other frames.

[0096] Advantageously said p-frame being less important in priority than key-frame, and more important in priority than b-frame.

[0097] Advantageously said b-frame being less important in priority than p-frame.

[0098] Advantageously said b-frame being less important in priority than key-frame.

[0099] Advantageously said unknown-frame being either more or less important in priority than key-frame, p-frame and b-frame.

[0100] Advantageously said key-frame being more important in priority than p-frames and b-frames other frames.

[0101] According to another aspect of the invention a machine-readable medium that provides executable instructions, which when executed by a set of processors, cause said set of processors to perform negotiating operations for various types of streaming media data by a server comprises:

receiving a request for one or more types of streaming media data from a caching proxy server or a client, said request including a request for data associated with said streaming media data, said request including an identifier which represents one of sev-

Google Exhibit No. 1008
Page 350

eral possible types of data associated with said streaming media data;
determining if requested types of streaming media data are supported by the server; and
responding to the request with a response indicating the capability of the server to support the request.

[0102] Advantageously said machine-readable medium further comprises receiving a request to send supported RTP extensions to the caching proxy or the client.
[0103] Advantageously said machine-readable medium further comprises responding to request to send and sending all the supported and requested extensions.
[0104] Advantageously said machine-readable medium further comprises receiving a command terminating the negotiation process.
[0105] According to another aspect of the invention a machine-readable medium that provides executable instructions, which when executed by a set of processors, cause said set of processors to perform negotiating operations for various types of streaming media data by a caching proxy server comprises:

sending a request for one or more types of related or unrelated streaming media data to a server, said request including a request for data associated with said streaming media data, said request including an identifier which represents one of several possible types of data associated with said streaming media data;
receiving a response to each requested type of streaming media data; and
deciding whether to proceed or terminate negotiation process associated with streaming media data.

[0106] Advantageously said deciding comprises:

determining if any requested type of streaming media data is not supported by server;
checking if unsupported type of streaming media data is essential to caching proxy server operations; and
sending an execution command to server.

[0107] Advantageously determining supported types of streaming media data being performed by checking if a response in a form of an echo or otherwise has been sent for requested type of streaming media data.
[0108] Advantageously said execution command being send based on results of checking if unsupported type of streaming media data is essential to caching proxy server operations.
[0109] Advantageously said decision being to terminate negotiation process.
[0110] Advantageously said decision being to proceed with negotiation process and requesting the server to send remaining supported type of streaming media data.
[0111] According to another aspect of the invention a machine-readable medium that provides executable instructions, which when executed by a set of processors, cause said set of processors to perform frame thinning operations by a caching proxy server comprises:

receiving a message from a client to thin frames in a streaming media data transmission from said caching proxy server;
evaluating priority of frames; and
sending only selected frames.

[0112] Advantageously said frame priority being evaluated by naming unsigned integers to frame types, said unsigned integers comprising:

assigning an integer value of "0" to an unknown frame type; an integer value of "1" to a key frame type; an integer value of "2" to a p-frame type; and an integer value of "3" to a b-frame type.

[0113] Advantageously said key-frame being most important in priority than any other frames.
[0114] Advantageously said p-frame being less important in priority than key-frame, and more important in priority than b-frame.
[0115] Advantageously said b-frame being less important in priority than p-frame.
[0116] Advantageously said b-frame being less important in priority than key-frame.
[0117] Advantageously said unknown-frame being either more or less important in priority than key-frame, p-frame a b-frame.
[0118] Advantageously said key-frame being more important in priority than p-frames, b-frames, and any other frames.
[0119] Advantageously said machine-readable medium further comprises:

receiving a second request from a client to further thin frames;
processing request and eliminating more selected frames and sending frames of higher priority.

[0120] Advantageously said selected frame being eliminated being a p-frame, and sending frames with higher priority than p-frame to the client.
[0121] Advantageously said selected frame being eliminated being a p-frames and b-frame, and sending frames with higher priority than both p-frames and b-frames to the client.
[0122] According to another aspect of the invention a machine-readable medium that provides executable instructions, which when executed by a set of processors, cause said set of processors to perform frame thinning operations by a client comprises:

Google Exhibit No. 1008
Page 351

sending a thinning message to a caching proxy server indicating not to send low order frames;
receiving frames back from caching proxy server that are higher in order than low order frames.

[0123] Advantageously the machine-readable medium further comprises:

sending a subsequent message from a client to further thin frames;
receiving frames that are higher in order then previous frames received.

[0124] Advantageously said frames being assigned unsigned integers, said unsigned integers comprising:

assigning an integer value of "0" to an unknown frame type; an integer value of "1" to a key frame type; an integer value of "2" to a p-frame type; and an integer value of "3" to a b-frame type.

[0125] Advantageously said key-frame being most important in priority than any other frames.

[0126] Advantageously said p-frame being less important in priority than key-frame, and more important in priority than b-frame.

[0127] Advantageously said b-frame being less important in priority than p-frame.

[0128] Advantageously said b-frame being less important in priority than key-frame.

[0129] Advantageously said unknown-frame being either more or less important in priority than key-frame, p-frame and b-frame.

[0130] Advantageously said key-frame being more important in priority than p-frames, b-frames and any other frames.

[0131] Advantageously said eliminating comprising eliminating p-frames and sending selected frames that are higher in order than p-frames to the client.

[0132] Advantageously said eliminating comprising eliminating both p-frames and b-frames, and sending selected frames that are higher in order than both p-frames and b-frames.

[0133] According to another aspect of the invention, a method of producing a representation of a streaming media data at a caching proxy server is provided, said method comprising:

transmitting a request for streaming media data to be delivered to said caching proxy server;
transmitting a request for data associated with said streaming media data, said request including an identifier which represents one of several possible types of data associated with said streaming media data;
receiving said streaming media data and storing said streaming media data on a storage device which is capable of being controlled by said caching proxy server; and
receiving said data associated with said streaming media data.

[0134] Advantageously, the method further comprises:

storing said data associated with said streaming media data in said storage device.

[0135] According to another aspect of the invention, a method for data transmission from a server data processing system is provided, said method comprising:

receiving a request for streaming media data, said request including a request for data associated with said streaming media data, said request including an identifier which represents one of several possible types of data associated with said streaming media data;
responding to the request with a response indicating a capability of the server to support the request; and
sending the requested data associated with said streaming media data.

[0136] Advantageously, said sending uses a real-time transport protocol (RTP).

[0137] Advantageously, said request may be made by a caching proxy server or a client.

[0138] Advantageously, the server responding with an echo only if it supports the request.

[0139] Advantageously, the method further comprises sending the requested data associated with the transmission protocol in an extensible extended header format.

[0140] Advantageously, the extensible extended header comprises an extension name and an extension identification (ID) associated with each separate RTP extension.

[0141] Advantageously, request may be for one or more type of transmission protocol data at a time.

[0142] Advantageously, the response by the server comprising response for each supported transmission protocol data and no response for any unsupported transmission protocol data.

[0143] Advantageously, the method further comprises receiving a request to send the transmission protocol data after sending a response for supported data, and sending only the requested and supported transmission protocol data.

[0144] According to another aspect of the invention, a machine-readable medium that provides executable instructions is provided, which when executed by a set of processors, cause said set of processors to send transmit time of RTP packet transmission operations from a caching proxy comprising:

Google Exhibit No. 1008
Page 352

requesting data corresponding to transmit time for streaming media data from a server;
receiving said streaming media data corresponding with transmit time information from the server;
storing the received information; and
transmitting from said caching proxy server to a client, said streaming media data at times specified by said transmit time.

[0145] According to another aspect of the invention, a caching proxy server is provided comprising:

means for transmitting a request for streaming media data to be delivered to said caching proxy server;
means for transmitting a request for data associated with said streaming media data, said request including an identifier which represents one of several possible types of data associated with said streaming media data;
means for receiving said streaming media data and storing said streaming media data on a storage device which is capable of being controlled by said caching proxy server; and
means for receiving said data associated with said streaming media data.

[0146] According to another aspect of the invention, a server data processing system is provided comprising:

means for receiving a request for streaming media data, said request including a request for data associated with said streaming media data, said request including an identifier which represents one of several possible types of data associated with said streaming media data;
means for responding to the request with a response indicating a capability of the server to support the request; and
means for sending the requested data associated with said streaming media data.

[0147] According to another aspect of the invention, a caching proxy server is provided comprising:

means for sending a message for streaming media data to a server, said request including a request for data associated with said streaming media data, said request including an identifier which represents one of several possible types of data associated with said streaming media data;
means for receiving a response from the server indicating support for the requested streaming media data;
means for informing the server to send the supported data associated with said streaming media data;
means for receiving the streaming media data from the server;
means for receiving a request from the client to send streaming media data; and
means for sending the requested streaming media data to the client.

[0148] According to another aspect of the invention, a RTP header is provided comprising:

means for adding a first RTP sub-extension ID to an RTP header;
means for defining a length of said first RTP sub-extension by providing a sub-extension length;
means for providing data corresponding to the RTP sub-extension ID within said length defined for said first RTP sub-extension; and
means for having subsequent RTP sub-extensions following the first RTP sub-extension.

[0149] According to another aspect of the invention, a server comprises:

means for receiving a request for one or more types of streaming media data from a caching proxy server or a client, said request including a request for data associated with said streaming media data, said request including an identifier which represents one of several possible types of data associated with said streaming media data;
means for determining if requested types of streaming media data are supported by the server; and
means for responding to the request with a response indicating the capability of the server to support the request.

[0150] According to another aspect of the invention,a caching proxy server comprises:

means for sending a request for one or more types of related or unrelated streaming media data to a server, said request including a request for data associated with said streaming media data, said request including an identifier which represents one of several possible types of data associated with said streaming media data;
means for receiving a response to each requested type of streaming media data; and
means for deciding whether to proceed or terminate negotiation process associated with streaming media data.

[0151] According to another aspect of the invention, a frame thinning caching proxy server comprises:

means for receiving a message from a client to thin frames in a streaming media data transmission from said caching proxy server;
means for evaluating priority of frames; and
means for sending only selected frames.

Google Exhibit No. 1008
Page 353

[0152] According to another aspect of the invention, a client comprises:

means for sending a message to a caching proxy server, said message indicating a need to thin streaming media data received at said client;
means for receiving media back from caching proxy server that are higher in order than low order media.

[0153] According to another aspect of the invention, a caching proxy server comprises:

means for requesting data corresponding to transmit time for streaming media data from a server;
means for receiving said streaming media data and corresponding transmit time information from the server;
means for storing the received information; and
means for transmitting from said caching proxy server to a client said streaming media data at times specified by said transmit time.

[0154] Other features and advantages of the present invention will be apparent from the accompanying drawings, and from the detailed description, which follows below.

BRIEF DESCRIPTION OF THE DRAWINGS

[0155] The present invention is illustrated by way of example and not limited by the figures of the accompanying drawings in which like references indicate similar elements and in which:

Figure 1a is a flowchart which shows a method in the prior art for transferring streaming media data to caching proxy server and then to a client.
Figure 1b illustrates a network of computer systems in which media data may be exchanged and/or processed, according to one embodiment of the present invention.
Figure 2 illustrates a block diagram of an exemplary digital processing system, which may be used in accordance with one embodiment of the present invention.
Figure 3 illustrates one embodiment of a communication method between a server and a client using RTSP and RTP protocols.
Figure 4 illustrates another embodiment of a communication method between a server, caching proxy server and a client.
Figure 5 illustrates one embodiment of a RTSP, RTP negotiation process between a caching proxy and a server.
Figure 6 illustrates one embodiment of a relationship between the server, caching proxy, and client during a transfer of a Transmit Time (TT) sub-extension to the caching proxy server and its use of TT information in transmitting streaming data to a client.
Figure 7 illustrates one embodiment of process that takes place during transfer of a transmit time sub-extension between server and caching proxy server.
Figure 8 illustrates one embodiment of process that takes place during transfer of a frame type sub-extension between server, and caching proxy server.
Figure 9 is a flow diagram of one embodiment of an operation to provide various types of information to a caching proxy in an extensible header format.
Figure 10 illustrates one embodiment of a relationship between the server, caching proxy, and client during a transfer of a Frame Type sub-extension.
Figure 11 illustrates a block diagram of a machine readable medium which stores executable computer program instruction for execution by an exemplary caching proxy server, which may be used in accordance with one embodiment of the present invention.
Figure 12 illustrates a block diagram of a machine readable medium which stores executable computer program instruction for execution by an exemplary originating server (server), which may be used in accordance with one embodiment of the present invention
Figure 13 illustrates a block diagram of a machine readable medium which stores executable computer program instruction for execution by an exemplary client, which may be used in accordance with one embodiment of the present invention.

DETAILED DESCRIPTION

[0156] A method and system for providing multimedia data transmission using real-time transfer protocol (RTP) and real time streaming protocol (RTSP) are described. For purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. For example, various computer network system architectures and digital processing system architectures are provided for illustrative purposes rather than to be construed as limitations of the present invention. It will be evident, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form to facilitate explanation.

[0157] **Figure 1b** is a diagram of a network of computer systems in which media data may be processed, according to one embodiment of the present invention. As shown in Figure 1b, a number of client computer system, one or more of which may represent one implementation of a receiving system, are coupled together through an Internet 122. It will be appreciated that the term "Internet" refers to a network of networks. Such networks may use a variety of protocols for exchange of information,

Google Exhibit No. 1008
Page 354

such as TCP/IP, ATM, SNA, SDI, RTP, RTSP etc. The physical connections of the Internet and the protocols and communication procedures of the Internet are well known to those in the art. Access to the Internet 103 is typically provided by Internet service providers (ISPs), such as the ISP 124 and the ISP 126, which may also be connected with caching proxy servers 130 and 132. Users on client systems, such as the client computer systems 102, 104, 118, and 120, generally obtain access to the Internet through Internet service providers, such as ISPs 124 and 126, which may also be connected through the internet with caching proxy servers 130 and 132. Access to the Internet may facilitate transfer of information (e.g., email, text files, media files, etc.) between two or more digital processing systems, such as the client computer systems 102, 104, 118, and 120 and/or a streaming media server system 128 which may be considered an originating server from which caching proxy servers receive streaming media data. For example, one or more of the client computer systems 102, 104, 118, and 120 and/or the streaming media server 128 may provide media data (e.g., video and audio, or video, or audio) to another one or more of the client computer systems 102, 104, 118, and 120 and/or the streaming media server 128. Such may be provided in response to a request. As described herein, such media data may be transferred in the system 100 according tracks. Such tracks, in one embodiment of the invention, may be created according to a specific format of the streaming media data and/or a specific data communication (e.g., network) protocol(s).

[0158] The streaming media server 128 is typically comprised of at least one computer system to operate with one or more data communication protocols, such as the protocols of the World Wide Web, and as such, is typically coupled to the Internet 122. Optionally, the streaming media server 128 may be part of an ISP which may provide access to the Internet and/or other network for client computer systems. The client computer systems 102, 104, 118, and 120 may each, with appropriate web browsing software, access data, such as HTML documents (e.g., Web pages), which may be provided by the streaming media server 128. Such data may provide media, such as QuickTime movies or QuickTime streaming media data, which may be presented by the client computer systems 102, 104, 118, and 120.

[0159] The ISP 124 provides Internet connectivity to the client computer system 102 via a modem interface 106, which may be considered as part of the client computer system 102. The client computer system may be a conventional computer system, such as a Macintosh computer, a "network" computer, a handheld/portable computer, a Web TV system, or other types of digital processing systems (e.g., a cellular telephone having digital processing capabilities). Similarly, the ISP 126 provides Internet connectivity for the client computer systems 104, 118 and 120, although as depicted in Figure 1b, such connectivity may vary between various client computer systems, such as the client computer systems 102, 104, 118, and 120. For example, as shown in Figure 1b, the client computer system 104 is coupled to the ISP 126 through a modem interface 108, while the client computer systems 118 and 120 are part of a Local Area Network (LAN). The interfaces 106 and 108, shown as modems 106 and 108, respectively, in Figure 1b, may be an analog modem, an ISDN modem, a cable modem, a satellite transmission interface (e.g., "Direct PC"), a wireless interface, or other interface for coupling a digital processing system, such as a client computer system, to another digital processing system. The client computer systems 118 and 120 are coupled to a LAN bus 112 through network interfaces 114 and 116, respectively. The network interfaces 114 and 116 may be an Ethernet-type, Asynchronous Transfer Mode (ATM), or other type of network interface. The LAN bus is also coupled to a gateway digital processing system 110, which may provide firewall and other Internet-related services for a LAN. The gateway digital processing system 110, in turn, is coupled to the ISP 126 to provide Internet connectivity to the client computer systems 118 and 120. The gateway digital processing system 110 may, for example, include a conventional server computer system. Similarly, the streaming media server 128 may, for example, include a conventional server computer system.

[0160] The system 100 may allow one or more of the client computer systems 102, 104, 118, and 120 and/or the streaming media server 128 to provide media data (e.g., video and audio, or video, or audio) to another one or more of the client computer systems 102, 104, 118, and 120 and/or the streaming media server 128. Such data may be provided, for example, in response to a request by a receiving system, which may be, for example, one or more of the client computer systems 102, 104, 118, and 120.

[0161] **Figure 2** is a block diagram of an exemplary digital processing system which may be used in accordance with one embodiment of the present invention. For example, the digital processing system 250 shown in Figure 2 may be used as a client computer system, a streaming media server system, a conventional server system, etc. Furthermore, the digital processing system 250 may be used to perform one or more functions of an Internet service provider, such as the ISP 124 or 126. The digital processing system 250 maybe interfaced to external systems through a modem or network interface 268. It will be appreciated that the modem or network interface 268 may be considered as part of the digital processing system 250. The modem or network interface 168 may be an analog modem, an ISDN modem, a cable modem, a token ring interface, a satellite transmission interface, a wireless interface, or other interface (s) for providing a data communication link between two or more digital processing systems.

[0162] The digital processing system 250 includes a processor 252, which may represent one or more proc-

Google Exhibit No. 1008
Page 355

essors and may include one or more conventional types of such processors, such as a Motorola PowerPC processor, an Intel Pentium (or x86) processor, etc. A memory 255 is coupled to the processor 252 by a bus 256. The memory 255 may be a dynamic random access memory (DRAM) and/or may include static RAM (SRAM). The processor may also be coupled to other types of storage areas/memories (e.g., cache, Flash memory, disk, etc.), which could be considered as part of the memory 255 or separate from the memory 255.

[0163] The bus 256 further couples the processor 252 to a display controller 258, a mass memory 262, the modem or network interface 268, and an input/output (I/O) controller 264. The mass memory 262 may represent a magnetic, optical, magneto-optical, tape, and/or other type of machine-readable medium/device for storing information. For example, the mass memory 262 may represent a hard disk, a read-only or writeable optical CD, etc. The display controller 258 controls in a conventional manner a display 260, which may represent a cathode ray tube (CRT) display, a liquid crystal display (LCD), a plasma display, or other type of display device. The I/O controller 264 controls I/O device(s) 266, which may include one or more keyboards, mouse/trackball or other pointing devices, magnetic and/or optical disk drives, printers, scanners, digital cameras, microphones, etc.

[0164] It will be appreciated that the digital processing system 250 represents only one example of a system, which may have many different configurations and architectures, and which may be employed with the present invention. For example, Macintosh and Intel systems often have multiple busses, such as a peripheral bus, a dedicated cache bus, etc. On the other hand, a network computer, which may be used as a digital processing device of the present invention, may not include, for example, a hard disk or other mass storage device, but may receive routines and/or data from a network connection, such as the modem or interface 268, to be processed by the processor 252. Similarly, a Web TV system, which is known in the art, may be considered to be a digital processing system of the present invention, but such a system may not include one or more I/O devices, such as those described above with reference to I/O device(s) 266. Additionally, a portable communication and data processing system, which may employ a cellular telephone and/or paging capabilities, may be considered a digital processing system which may be used with the present invention.

[0165] In the system 250 shown in Figure 2, the mass memory 262 (and/or the memory 254) may store media (e.g., video, audio, movies, etc.) which may be processed according the present invention (e.g. by way of tracks). Alternatively, media data may be received by the digital processing system 250, for example, via the modem or network interface 268, and stored and/or presented by the display 260 and/or I/O device(s) 266. In one embodiment, packetized media data may be transmitted across a data communication network, such as a LAN and/or the Internet, in accordance with tracks. On the other hand, the processor 252 may execute one or more routines to use a file with one or more tracks, or alternatively, to create one or more tracks, to process media (e.g., a pre-packaged movie, audio file, video file, etc.) for presentation or packetization according to the tracks. Such routines may be stored in the mass memory 262, the memory 264, and/or another machine-readable medium accessible by the digital processing system 250. In one embodiment, the digital processing system 250 may process media data having tracks embedded therein. Similarly, such embedded media data may be stored in the mass memory 262, the memory 264, and/or another machine-readable medium accessible by the digital processing system 250.

[0166] **Figure 3** shows an example of components involved in data transmission scenario. An originating server 301 and a client 302 are shown as components involved in carrying out transmission of streaming media data using RTP and RTSP protocols as one embodiment of the present invention. The originating server 301 and the client 302 may communicate directly with each other or may communicate through an intermediary such as a caching proxy server. In one embodiment, the server 301 and the client 302 may be on separate local area networks (LAN). In another embodiment the server 301 and the client 302 may be connected through a wide area network. There may be either one or several clients 302 that are in communication with the server 301 directly or indirectly through an intermediary, such as the Internet. The server 301 and client 302 may interact with each other for sending various types of streaming media data in various formats. In one embodiment, the streaming media data may be sent in a downstream direction from server 301 to client 302. In another embodiment the client 302 may send requests and other streaming media data information to server 301.

[0167] **Figure 4** shows an example of one embodiment of a communication relationship between a client 302, a caching proxy server (CP) 401 and the originating server 301. There may be several types of connections between these components, but preferably the client 302 may be in communication with the caching proxy server 401 through an Internet connection, and the caching proxy server 401 may be in communication with the originating server 301 through an Internet connection

[0168] A caching proxy server 401 may be connected through the Internet with a single client 302 or several clients 302. The caching proxy server 401 and its connected clients 302 may be on the same local area network or may be connected through a wide area network. In one embodiment it is preferable that the caching proxy server 401 and client 302 or clients 302 are connected through a local area network and in close proximity to each other. An exemplary embodiment of close proximity connection may be connection in the same company etc. where the connection may utilize a high

Google Exhibit No. 1008
Page 356

bandwidth interface. The communicational link between the caching proxy server 401 and client 302 may be of a variety of types such as direct cable, fiber optic, radio frequency etc. These links may change and vary based on the need of a particular client 302 and advancements in technology.

[0169] A originating server 301 and a caching proxy server 401 may communicate using a communicational link such as direct cable, fiber optic, radio frequency etc. These links may change and vary based on a particular need and advancements in technology. The cashing proxy 401 may act as an intermediary between the originating server 301 and client 302 to transfer streaming media data and assist in smooth delivery of RTP packets from server 301 to client 302. In so doing, a caching proxy server 401 may perform several of its own functions. In one embodiment the caching proxy 401 functions may be thinning frames, storing streaming media data locally, and transmitting streaming media data at offset times to client 302. In another embodiment the caching proxy server's 401 functions may be negotiating with originating sever 301 for various RTP extension associated with various types of streaming media data, and receiving or responding to various client 302 requests etc. In one embodiment, one of the objectives of a caching proxy server 401 is to deliver a pristine and good quality copy of streaming media data to the client 302 and do so in an efficient and speedy manner.

[0170] Typically a client 302 may sent a request directly to the caching proxy server 401. The caching proxy server 401 may then react to the client 302 request and either fetch the requested items from the system server or responds on its own. Its own response may be from a copy of streaming media data which has already been obtained from an originating server and which has been stored on a storage device controlled by the caching proxy server (e.g. a local hard disk of the caching proxy server). However the system may also be configured for the client 302 to send requests directly to the system server 301 and have the server 301 respond back directly to the client 302 or indirectly to the client 302 through a caching proxy server 401.

[0171] **Figure 5** shows one exemplary method according to an embodiment of the present invention. In the operations of Figure 5, an originating server(e.g. server 301) and a caching proxy server 401 communicate with each other to assist in smooth transmission of streaming media data. This communication aids smooth packet delivery in many ways including allowing the caching proxy server 401 to deliver to the client 302 good quality streaming media data at a high speed. In addition, the communication also aids in assisting and managing client's load by ensuring that the client 302 gets a manageable amount of streaming media data and no frames are dropped in the process or only less important frames dropped in the process (through frame thinning).

[0172] Initially in operation 501, the caching proxy server requests streaming media data from an originating server. The request may be made by asking the server 301 for "setup" in RTSP for audio or video streaming media data. The request may be for one type of streaming media data or several types of streaming media. The request may be for similar or unrelated types of streaming media data. The server 301 receives the request from the caching proxy server 401, and the server 301 responds in the manner described with respect to operation 502 of Figure 5. The "SETUP" request in RTSP in operation 501 may be initiated by the caching proxy server 401, independently of a client system 302 requesting streaming media data or the request in operation 501 may be initiated by a client system 302 requesting the streaming media data from the caching proxy server 401 which in turn requests the requested streaming media data from the server 301 (if the caching proxy server 401 does not already have the requested streaming media data stored under its control, such as a local hard disk of the caching proxy server 401). The caching proxy server 401 may also log client's IP address for subsequent communication in the case where a client initiated the request.

[0173] The caching proxy server 401 and originating server 301 may establish a communication process in which the caching proxy server 401 and the originating server 301 may engage in a negotiation process 502 for communicating back and forth in order to aid a smooth streaming media data packet transmission. As shown in operation 501, the caching proxy server 401 may communicate with the originating server 301 and request (e. g. by specifying names of RTP extensions) a set of RTP extensions associated with the streaming media data to be sent to the caching proxy server 401. The set of extensions requested to the server 301 may be the same as the set of requests sent to the caching proxy server 401 from the client 302 (in those cases where the client specifies RTP extensions, such as security extensions, for its use).

[0174] The server 301 receives the request for RTP extensions from the caching proxy server 401. The server 301 may then run its internal processes to determine whether the server 301 supports the requested RTP extensions. The outcome of this determination may be that the server 301 supports some but not all the requested RTP extensions, or that the server 301 supports none of the requested RTP extensions, or that the server 301 supports all of the requested RTP extensions. The server 301 may respond in operation 502 to the caching proxy server 401 by informing the caching proxy server 401 of the server's 301 supported RTP extensions. The server 301 may choose to respond 502 by indicating only the supported RTP extensions or may respond by indicating both the supported and unsupported RTP extensions, or the server 301 may not respond at all indicating no support for requested extensions. In one embodiment the response may be in an echo form or any several other forms. In one echo form of the invention,

Google Exhibit No. 1008
Page 357

the server transmits the names of the requested RTP extensions and an associated code for each named extension.

[0175] The caching proxy server 401 receives a response from the server 301 indicating the supported RTP extensions or both the supported and unsupported RTP extensions: The caching proxy server 401 may check to see if a response has been sent for all the RTP extensions it had earlier requested. Caching proxy server 401 may have received none, one, some, or all responses to the requested RTP extensions. Caching proxy server 401 may evaluate further to check if any of the server 301 unsupported RTP extensions are required for streaming media data transmitting process. Required RTP extensions may be defined as RTP extensions that are necessary for carrying on a particular data transmission operation such as frame thinning etc at the caching proxy server 401. As shown in Figure 5, operations 501 and 502 relate to setup and negotiation for an audio track while operations 503 and 504 relate to similar setup and negotiation for a video / image track.

[0176] In one embodiment, the caching proxy server 401 may request multiple sets of RTP extensions at a time from the server 301. If the RTP extensions requested are required and unsupported by the server 301, then caching proxy server 401 may decide to terminate the negotiation process. It may also be the case that some of the extensions are supported and some are not. In such a situation, if the unsupported extensions are not required for the data transmission process then caching proxy server 401 may decide to proceed further and receive the supported extensions and the associated streaming media data. In another embodiment the caching proxy 402 may not receive a response for any of the RTP extensions requested. In such a case the caching proxy 402 may choose to terminate the negotiation process with the server 301.

[0177] If the caching proxy server 402 decides not to terminate the negotiation process and to request the supported RTP extensions and streaming media data, it may send a request to the server 301 to send the streaming media data and the associated supported RTP extensions in operation 504. In the example of Figure 5, this request for the streaming media data and the associated RTP extensions occurs when the caching proxy server 401 sends a "PLAY" command in the RTSP protocol.

[0178] The server 301 in operation 505, responds to the "PLAY" command by sending the streaming media data and by sending the requested and supported RTP extensions, which is associated with the streaming media data, to the caching proxy server 401 in a extended header format. This header may contain one, two or three similar or unrelated RTP extensions.

[0179] Upon receiving the streaming media data and receiving RTP extensions from the server 301, the caching proxy 401 may store the streaming media data and the RTP extensions in a storing facility 601 (e.g. a storage device controlled by the caching proxy sever 401, such as a local hard disk of the server 401) and terminate the transmission process with the server 301. The caching proxy 401 may again reinitiate the negotiation process and repeat all the back and forth if another request for streaming media is submitted by the client 302. This request may be similar or completely different from prior requests. Some of the extensions that may be requested by the cashing proxy sever 401 may be a transmit time sub-extension denoted by symbol "trti", or frame type sub-extension denoted by symbol "ftry", or packet position sub-extension denoted by symbol "papo". Other extensions may also be requested (e.g. an extension which is used by the client 302 or server 401 to maintain a secure or encrypted or authenticated communication between client 302 and server 401).

[0180] For example, in one cycle of its operation a caching proxy server 401 may ask for three separate RTP sub-extensions one of which may be frame type sub-extension denoted by symbol "frty" (used in frame thinning by caching proxy server 401 as described below), the other may be transmit type sub-extension denoted by "trti" (used by the caching proxy server 401 as described below), and the last may be packet position sub-extension denoted by "papo" (which may be used to retrieve lost or missing packets). Let us also assume for the illustration of this example that "frty" sub-extension is required for the streaming media data transmission process. "Frty" may be denoted as a required sub-extension due to several reasons. One of the reasons may be that the client 302 cannot receive or process the data at a high data rate (and so frame thinning is required) and "frty" sub-extension will assist the data transmission process between a caching proxy server and the client 302 by allowing the caching proxy sever to perform frame thinning and therefore may be "necessary".

[0181] The caching proxy 402 may receive the request and communicate with the server 301 by sending a single request to the server 301 asking for both sub-extensions. Let us assume further for the illustration of this example that the server 301 can only support one of the two RTP extensions. The server 301 may then send a response back to the caching proxy server indicating which sub-extension is supported.

[0182] If the supported sub-extension happens to be only "trti", or "papo" or both but not "frty" then the caching proxy 402 will terminate the negotiation process between the caching proxy 402 and the server 301. This is because "frty" was a required extension to the data transmission process and since it is not supported by the server 301, the caching proxy 402 may not proceed further. If however, the supported sub-extension happens to be only "frty", or frty and papo, or frty and trti, or frty, papo and trti, then the caching proxy server 401 may proceed further with the transmission process. The caching proxy server 401 in this instance may choose not to terminate the process since the required sub-ex-

Google Exhibit No. 1008
Page 358

tension frty is present in the response as supported by the server 301.

[0183] **Figure 6** shows an example of a method for transmitting packet transmit time data which may be used with various embodiments of the present invention. The server 301 is connected with the caching proxy server 401 by way of a standard communication carrying devices such as fiber optic wire link, radio frequency communication, cable wire etc. A person having ordinary skill in the art will appreciate that any one-communication device is not essential for the data transfer operation in accordance with this invention and that these communications devices are interchangeable. It must be clear that it is important for the communication devices to allow communication in both directions i.e. from server 301 to caching proxy 402 or from caching proxy 402 to server 301.

[0184] The communication between a caching proxy server 401 and the originating server 301 may be a direct communication relationship or there may also be other devices such as routers in the Internet acting as intermediaries to assist in streaming media data transfer. Typically, a caching proxy server 401 is located in closer proximity to the client 302 than the originating server 301. This close proximity may be within a company, or on a designed local area network (LAN), or in the same geographic region, whereas typically caching proxy server and original system server 301 are further apart.

[0185] The caching proxy server 401 may have a storage facility 601 to store streaming media data 603 and / or the associated RTP extensions 602. The storage facility 601 may be a local to the caching proxy server 401 or on an offsite from the caching proxy server 401 but in either case the storage is controlled by the caching proxy server 401. The caching proxy server 401 may have a link established to store data received from the server 301 for a periods of time in the storage facility 601, and then be able to retrieve the stored data at a later time for sending to client 302. In the example of Figure 6, the streaming media data 603 and its associated RTP extension (transmit time in this case) are stored together on a storage device 601. Groups of streaming media data (e.g. a packet or a set of packets) are associated with a corresponding designation of a transmit time so that each group has a transmit time which specifies when to transmit the particular group. It will be appreciated that the streaming media data and the associated RTP extension may be stored separately (but still be associated -- e.g. packet No. xxx to be transmitted at time ABC, packet No. xxy is to be transmitted at time ABD, etc.).

[0186] In the example of Figure 6, the streaming media data is received by the server 401 and the caching proxy server 401 receives the transmit time data from server 301 and stores it in the storing facility 601. Transmit time data may be associated with each track of streaming media data. For example, in one instance the transmit time at 0 sec 602 may be associated with corresponding streaming media data 603. In operation, in this exemplary embodiment, the streaming media data 0 will be sent to a client at transmit time 0.

[0187] **Figure 7** shows one exemplary method for using transmit time as an RTP extension according to an embodiment of the present invention. In operation the method suggested in Figure 7 may utilize the system architecture as suggested in one of the embodiments of the present invention shown in Figure 6.

[0188] In one example of the method of Figure 7, a caching proxy server 401 receives a request from client 302 for streaming media data and then requests an RTP extension which specifies transmit time information and requests the server 301 to send transmit time sub-extension RTP data 701 and associated streaming media data. Operation 701 shows the caching proxy server's request for streaming media data and transmit time which results from this request. The server receives the request in operation 702 as shown in Figure 7. It may also be the case that a caching proxy server 401 already had received the requested streaming media data and its associated transmit time information from the server 301 and has stored the streaming media data and associated RTP extensions at a storing facility 601. If such, then the caching proxy server 401 may start responding to clients 302 request without communicating with the originating server 301 thereby shipping to operations 707 and 708 of Figure 7.

[0189] Assuming for illustration of this example that the original server 301 supports the transmit time information, server 301 will respond back to caching proxy server indicating its support of the requested sub-extension in operation 703. If however the transmit time sub-extension is not supported by the original server 301, the originating server 301 may or may not respond back to the caching proxy server 401 indicating its support for the requested sub-extension as shown in operation 709. In the event of an unsupported sub-extension, the caching proxy 402 may terminate the negotiation process as shown in operation 710 with the server 301 and would typically inform the client 302 of the inability to provide streaming media data. In so doing, the caching proxy server 401 may first evaluate whether the missing transmit time information is required for running its processes. If the result of the determination is that transmit time information in this particular example is a required element, then the caching proxy server may decide whether to proceed or terminate the transmission process.

[0190] The server 301 in operation 704 sends the transmit time RTP data in an extended header format according to the RTP protocol to the caching proxy server. The header may consist of the normal header fields, the sub-extension character name and a sub-extension ID 704. The sub-extension character name for a transmit time data may be a 4-character code denoted by "trti". This code may uniquely identify and describe the content of the sub-extension as being transit time data.

Google Exhibit No. 1008
Page 359

The sub-extension ID may identify the sub-extension in the RTP packet.

[0191] A transmit time sub-extension may consist of a single 64-bit unsigned integer representing the recommended transmission time of the RTP packet in milliseconds as shown in operation 704. The transmit time may be offset from one another from the start of a media presentation. For example in one sub-cycle of operation, a session description protocol document for a uniform resource locator (URL) may include a range of 0-729.45 seconds. The client 302 may make a PLAY request 706 for the video, audio, text, graphics, and images etc. type data.

[0192] The caching proxy server 401 may receive the RTP data packet associated with streaming media data with the transmit time sub-extension as shown in more detail in figure 6. The caching proxy server 401 may then store the RTP transmit time data locally as shown in Figure 6. The caching proxy server 401 may then strip off the header ID in operation 705 and send streaming media data associated with each track, in operation 707, of transmit time individually at offset times to the client 302 allowing the client 302 to carry on PLAY operation 708. An advantage of knowing and storing transit time at offsets locally at the caching proxy server, it may now be possible for the caching proxy server 401 to selectively re-transmit data at different intervals to the client 302 or respond to clients request to send data corresponding to any particular time slot.

[0193] **Figure 8** shows one exemplary method for a stream thinning process by a caching proxy server according to an embodiment of the present invention. In operation a client 302 and caching proxy server 301 communicate with each other to assist in sending and receiving streaming media data and assisting in traffic flow control to the client 302. In a method according to Figure 8, a client 302 communicates with the caching proxy server 401 and indicates that it is overloaded or the caching proxy server 401 detects that the client is overloaded. As part of this communication, the caching proxy server 401 ensures that the client 302 does not get an amount of data that exceeds its data handling capacity. Caching proxy server also prevents at least selected frame being "dropped" or missing as a result of an overloaded client 302.

[0194] A principle behind Figure 8 is that an overloaded client 302 may notify the caching proxy server that it has reached its capacity for receiving RTP data (e.g. streaming media data). The client 302 may have been overloaded due to several reasons including that a caching proxy server is sending RTP data very quickly and the client 302 is having difficulty receiving data at such a fast pace. The client 302 may inform the caching proxy server to stop sending streaming media data altogether, or to send data at a slower pace. The client 302 may also inform the caching proxy server to send only selected order of frames and not send any low order frames. The caching proxy server 401 will use the frame type data to determine which frames to transit to client 302; typically, higher priority frames are transmitted while lower priority frames are not transmitted.

[0195] A method of Figure 8 begins in operation 801 in which a caching proxy server 402 may communicate with the originating server 301 and request the server 301 for streaming media data and its associated frame type information. The frame type identifies various types of data (e.g. frames) in streaming media data which allows "thinning" which may be defined as reducing frames, sending frames at a slower pace, or not sending certain frames at all. It will be appreciated that thinning apples to various types of data and that "frames" may be considered to be such various types of data. The server 301 may receive the request in operation 802 and may respond in operation 803 to the caching proxy server 401 indicating whether the server 301 supports the requested frame type streaming media data. If the server 301 supports this, the server's 301 response in operation 803 includes sending the associated RTP frame type sub-extension in a format described in block 804 along with an identifier code corresponding to the frame type extension requested by name in operation 801.

[0196] If the server 301 does not support frame type sub-extension then the caching proxy server may terminate in operation 807 and 808 the communication with server 301. The server 301 may indicate that it does not support the requested frame type streaming media data by either responding or not sending any response to the Caching Proxy-server 401 which would also indicate no support of the requested RTP extension for the streaming media data. However, if the server 301 supports the frame type sub-extension, the caching proxy 402 may inform the server 301 to send the streaming media associated with the frame type information. In one embodiment, the server 301 may send the supported streaming media data sub-extensions without any further requests from the caching proxy server 401. In another embodiment, the server 301 may wait for further a further request from the caching proxy server 401 to send the supported streaming media data sub-extensions.

[0197] The server 301 may then send the RTP sub-extension in an extended header format. The frame type sub-extension may consist of a single 16-bit unsigned integer value with several well-known values representing different frame types. The well-known values may be "1" for a key frame, "2" for a p-frame, or "3" for a b-frame where key frame maybe of the highest order and most importance, b-frame of the lowest order and least importance, and b-frame somewhere between key frame and b-frame in terms of importance. There may also be other frames that may be added to this format.

[0198] The caching proxy server 401 may then store the streaming media data and its associated frame type sub-extension in its storing device 601 after receiving them from the originating server 301. This is shown in operation 805 of Figure 8. The caching proxy server 401 may then enter into a negotiating process with the client

Google Exhibit No. 1008
Page 360

302 in evaluating the client's capability at the time to handle streaming media data traffic 809. Based upon the result of the negotiation process 809, the caching proxy server 401 may thin frames (sending only selected, predetermined frames) and send streaming media data associated with selected frames 806 to the client 302.

[0199] For example, in one cycle of operation a client 302 may inform the caching proxy server that it is overloaded. The client 302 may inform the caching proxy server 401 to stop sending frames altogether or to lower the bit rate if the transmission falls behind. In the case of lowering the bit rate and slowing down, the caching proxy server 401 may stop sending the lowest order frames of the streaming media data, the b-frame to the client 302. The caching proxy server 401 and the client 302 may communicate further to evaluate if the client 302 is still overloaded. In one embodiment, if the client 302 is capable of handling the load after thinning of the b-frame then the caching proxy server may send the client 302 key-frames and p-frames. However if the client 302 is still overloaded then the caching proxy server 401 may further reduce the data traffic to the client 302 and stop sending p-frames. The caching proxy server 401 may further evaluate client's 302 data handling capability and determine if any more frame thinning is necessary to reduce load on client 302. In another embodiment the client 302 may directly specify to the caching proxy server 401, which frames to send and which frames not to send until a subsequent request is sent to the caching proxy server 401 to change sending considerations.

[0200] After a client 302 retains its capability to cache frames, the caching proxy server 401 may again start sending the lower order frames to the client 302. It may again send all the frames at a high speed or send the frames according to requests received by the client 302. In the event that the client 302 gets overloaded again, the caching proxy server 401 may repeat the thinning process until the client 302 is able to handle caching data again. Figure 10 shows an example of how a caching proxy server 401 receives streaming media data and its associated frame type (FT) RTP extension data from an originating server 301 and stores the streaming media data and associated frame type extension data on a storage device (e.g. a local hard disk of the caching proxy server 401) and then uses the frame type data to selectively thin frames of the streaming media data which is being transmitted to a client 302.

[0201] Communication between a caching proxy server 401 and originating server 301 or caching proxy server 401 and client 302 is carried on using real-time transfer protocol (RTP) and real-time streaming protocol (RTSP) for sending / receiving streaming media data. An originating server 301 sends streaming media data packets in a streaming media format using RTP to a caching proxy server 401 whenever a transmission of streaming media data occurs. One of the embodiments of the present invention is to be able to modify the current existing RTP headers by being able to expand the header with sub-extensions and also be able to make the header format variable. Expansion of the header is useful because a caching proxy server 401 may need several pieces of information along with a RTP packet that will aid in providing a good quality streaming media data packet and smooth delivery to the client 302. The extra information that may be needed can be provided by attaching it to the existing header by being able to expand the header field. It should also be clear that variability of the extended header is important because the extra pieces of information needed by the caching proxy 402 may vary each time. To accommodate for this variation, the extended header may have the capability to change and provide various types of information as needed by the caching proxy server 401.

[0202] In accordance with one embodiment of the invention, in operation, an extended header consists of a normal header fields. A person having ordinary skill in the art is aware of the various header fields that are normally used in operation. The normal header fields are immediately followed by header extension fields. The extension field consists of several sub-extensions. There may be several header sub-extensions that are unrelated to each other and may vary per request of the caching proxy server 401. The sub-extensions may have an extension type of "se". The RTP extension length may be the total length of all the sub-extensions and may be defined in 32-bit words thereby being in full compliance with the RTP protocol.

[0203] The "se" sub-extension format may be such that a sub-extension ID immediately follows the normal RTP header field. The ID may identify the sub-extension within the RTP packet. This ID may be a one octet ID generated by the server 301 for each individual named RTP sub-extension. Each sub-extension may also have its unique name that is defined by a four-character name code. This name code uniquely identifies and describes the type of data in each sub-extension. For example, the four character name code for a transmit time sub-extension may be "trti", frame type sub-extension may be "frty" and packet position sub-extension maybe "papo". This name code is associated with the one octet ID (generated by the server 301) so that the caching proxy server 401 can identify, form the octet ID the appropriate RTP extension data when it receives streaming media data.

[0204] In one embodiment of the present invention, the unique name may be "frty" associated with streaming media data for frame type information. The unique name "frty" may also have an unsigned integer associated with each different type of frame. In one embodiment the unsigned integer may be "1" for a key-frame, "2" for a p-frame, and "3" for a b-frame. A user may also add any additional frames in the future as need and technology advances and may use this header format without any need for much modifications.

Google Exhibit No. 1008
Page 361

[0205] In another embodiment of the present invention, the unique name may be "trti" associated with streaming media data for transmit time type information.

[0206] In another embodiment of the present invention, the unique name may be "papo" associated with streaming media data for packet position type information.

[0207] **Figure 9** shows an exemplary method of several aspects of the present invention. In a portion 901, a caching proxy server 401 requests streaming media data from an originating server 301 and also requests by name one or more RTP extensions. This request is made using the RTSP protocol. In operation 903, the server typically responds back (e.g. of a response would be an echo) a response to the caching proxy server 401 indicating its support for the requested RTP extensions. The server 301 also transmits to the caching proxy server 401 an identifier, such as a number code which corresponds to each name of the requested RTP extensions. Typically, the caching proxy server 401 will use the number code later in identifying received extended RTP data. The number code allows the caching proxy server 401 to identify the various types of RTP extension data in the streaming media which it receives as the server 301 may not use the name to designate the RTP extension type. In operation 905, the caching proxy server 401 receives the server's 301 response and then in operation 907, the CP server 401 determines whether the server 301 responded to all of the requested RTP extensions.

[0208] If the server 301 did not respond to all requested RTP extensions, then processing proceeds to operation 909, followed by operation 911 in which it is determined whether any of the missing RTP extensions are critical to the caching proxy server's 401 processing. If they are not critical, then processing proceeds to operation 921. If they are critical, then the caching proxy server 401 determines in operation 913 whether or not to terminate the operation/communication with the originating server 301. As shown in operations 915 or 917, the caching proxy server 401 may terminate operations/communications with the server 301 for this particular streaming media data which was requested or they proceed to receive the streaming media and whatever supported extensions can be provided.

[0209] In operation 921, the CP server 401 requests the originating server 301 to send the requested streaming media data and its associated RTP extensions. In one embodiment, the CP server 401 transmits a "PLAY" request using RTSP, and this causes the server 301 to respond in operation 923 by transmitting the streaming media data and the associated RTP extensions. In operation 925, the CP server 401 stores the streaming media data received from the server 301 and also stores the associated RTP extension data. In operation 927, the CP server 401 may remove certain RTP extension data from the streaming media file, such as the transmit time or the frame type data. This is done in order to avoid sending the transmit time or the frame type information to the client 302 which requests streaming media data. The RTP extension data, which is removed from the streaming media data, is stored separately but associated with the streaming media data. For example, transmit times for various packets are stored separately from the packets, but the association existing in the data received from the server 301 between the transmit time and the corresponding packets is maintained even when the transmit times are stored separately so that the caching proxy server 401 may determine the appropriate transmit time for each of the packets in the streaming media data. In operation 929, the caching proxy server 401 evaluates a client's 302 request for streaming media data and responds accordingly. It will be appreciated that a client 302 will negotiate for streaming media data using the RTSP protocol and the CP server 401 will respond with the streaming media data by transmitting the data to the client 302. In addition, the client 302 may request frame thinning. Further, the caching proxy server 401 may use the transmit times to determine when to transmit to various packets in the streaming media data to the client 302.

[0210] **Figure 11** shows one type of exemplary machine readable media (e.g. RAM or hard disk or combination thereof) for storing executable computer program instructions for a caching proxy server 401 that may be used in accordance with the present invention. The caching proxy server 401 typically will have its own operating system (OS) software 1101. This software 1101 may be the Macintosh OS. Or Windows NT or Unix, or other well known operating systems

[0211] The control software 1102 is for transmitting or receiving streaming media data using, for example RTP and RTSP protocols. The software 1102 is normally able to retrieve or send various types of streaming media data packets and direct commands for storing the received media in a storing facility 601. Thus software 1102 performs the negotiation process with an originating server 301 and receives streaming media data, and its associated RTP extensions and causes the streaming media data and its associated RTP extensions to be stored on a storage device controlled by caching proxy server 401. Figure 11 shows the storage of two streaming media data files 1103 and 1104.

[0212] Streaming media data file 1103 may contain streaming media data 1 in streaming media format 1105, transmit time associated with streaming media 1 (1106), and frame type associated with streaming media 1 (1107). In one embodiment, the operating system 1101 and control software 1102 may have the capability to separate streaming media data in packet 1 from other packets and store it separately in a storing facility 601 and to extract the RTP extensions (e.g. Transmit Time data or Frame Type data) from the stored streaming media packets and store these separately so that these packets do not include the RTP extensions.

[0213] Streaming media data file of 1104 may contain

Google Exhibit No. 1008
Page 362

streaming media data 2 in streaming media format 1108, transmit time associated with streaming media data 2 (1109), and frame type associated with streaming media 2 (1110).

[0214] The streaming media data 1105 and 1108 will usually not be in the same original format as the media data was at the originating server 301. The streaming media data 1105 and 1108 may however be a full "pristine" copy of the original media data, because the "papo" extension may be used by the caching proxy server 401 to search for any missing packets in the streaming media data 1105 and 1108 and to request (again) these packets from the originating server.

[0215] **Figure 12** shows one type of exemplary machine readable media (e.g. RAM or hard disk or combination thereof) for storing executable computer program instructions for an originating server 301 that may be used in accordance with the present invention. The server 301 will typically have its own operating system 1201.

[0216] The control software 1202 is for transmitting streaming media data to a caching proxy server 401 or to a client 302 using the RTP and RTSP protocols and the RTP extensions of the invention. Further, software 1202 receives requests from a client 302 or a caching proxy server 401 for streaming media and negotiates with a caching proxy server 401 for various types of streaming media data and associated RTP extensions, and responds to various requests by caching proxy servers 401 or clients 302.

[0217] Software 1204 converts original media data 1203, which is usually not in a packet format, to a streaming media data format (e.g. packet format) for transmitting to caching proxy sever 401 or client 302. When converted, the converted streaming media data is a representation of the original media data 1203 that has a different format than the format of the original media data 1203.

[0218] The software 1206 creates RTP extension headers associated with various types of streaming media data. The system may assign various ID names and codes 1205 associated with various RTP extensions to various types of streaming media data before its sent to a caching proxy 401 or a client 301. The software 1206, in conjunction with software 1202, performs the negotiation process with a caching proxy server 401 (or, in some cases where the client asks for an RTP extension, such as a security or encryption or authentication extension, the client) to transmit RTP extension data for an associated streaming media data and also performs the transmission process of transmitting streaming media data with its associated RTP extension.

[0219] **Figure 13** shows one type of exemplary machine readable media (e.g. RAM or hard disk or combination thereof) for storing executable computer program instructions for a client server 302 that may be used in accordance with the present invention. The client server 302 will typically have its own operating system 1301 such as a Macintosh OS, or Windows NT, or Unix, or other well-known operating systems. The client's media may also include Web Browser software 1303 such as Netscape's Navigator or Microsoft's Internet Explorer.

[0220] The streaming media data player software 1302 is for receiving and playing streaming media data transmitted to the client using the RTP protocol. The streaming media data player software 1302 may be QuickTime software from Apple computer or the Real Player from Real Networks. The streaming media data player software 1302 is typically able to send requests to a caching proxy server 401 or a server 301 for various different types of streaming media data and to receive and present (e.g. display images and produce sound) a representation of streaming media data.

[0221] In yet another embodiment the streaming media data player software 1302 may be able to communicate and negotiate with a caching proxy server 401 in order to regulate incoming data traffic to handle its load better (e.g. the software 1302 may ask a CP server 401 to perform frame thinning).

[0222] In the foregoing specification, the invention has been described with reference to specific exemplary embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather a restrictive sense.

Claims

1. A method for operating a caching proxy server comprising:

   sending a request for streaming media data to a server, said request including a request for data associated with said streaming media data, said request including an identifier which represents one of several possible types of data associated with said streaming media data;
   receiving a response from the server indicating support for the requested streaming media data;
   informing the server to send the supported data associated with said streaming media data;
   receiving the streaming media data from the server;
   receiving a request from the client to send streaming media data; and
   sending the requested streaming media data to the client.

2. A method for extending an RTP header comprising:

   adding a first RTP sub-extension ID to an RTP header;

Google Exhibit No. 1008
Page 363

defining a length of said first RTP sub-extension by providing a sub-extension length;
providing data corresponding to the RTP sub-extension ID within said length defined for said first RTP sub-extension; and
having subsequent RTP sub-extensions following the first RTP sub-extension.

3. A method of negotiating for various types of streaming media data by the server comprising:

receiving a request for one or more types of streaming media data from a caching proxy server or a client, said request including a request for data associated with said streaming media data, said request including an identifier which represents one of several possible types of data associated with said streaming media data;
determining if requested types of streaming media data are supported by the server; and
responding to the request with a response indicating the capability of the server to support the request.

4. A method of negotiating for various types of streaming media data by the caching proxy server comprising:

sending a request for one or more types of related or unrelated streaming media data to a server, said request including a request for data associated with said streaming media data, said request including an identifier which represents one of several possible types of data associated with said streaming media data;
receiving a response to each requested type of streaming media data; and
deciding whether to proceed or terminate negotiation process associated with streaming media data.

5. A method of frame thinning by caching proxy server comprising:

receiving a message from a client, said message indicating a need to thin streaming media data being sent to said client;
evaluating priority of streaming media data; and
sending only selected streaming media data.

6. A method of using transmit time of RTP packet transmissions at a caching proxy server said method comprising:

requesting data corresponding to transmit time for streaming media data from a server;
receiving said streaming media data and corresponding transmit time information from the server;
storing the received information; and
transmitting from said caching proxy server to a client said streaming media data at times specified by said transmit time.

7. A machine-readable medium that provides executable instructions, which when executed by a set of processors, cause said set of processors to perform data transmission receiving operations from a sever comprising:

sending a request for streaming media data to said server, said request including a request for data associated with said streaming media data, said request including an identifier which represents one of several possible types of data associated with said streaming media data;
receiving a response from said server indicating support for the requested streaming media data;
informing said server to send the supported data associated with said streaming media data;
receiving the supported streaming media data from said server;
receiving a request from a client to send streaming media data; and
sending the requested streaming media data to said client.

8. A machine-readable medium that provides executable instructions, which when executed by a set of processors, cause said set of processors to perform RTP header extending operations comprising:

adding a first RTP sub-extension ID to an RTP header;
defining a length of the first RTP sub-extension by providing a sub-extension length;
providing data corresponding to the RTP sub-extension ID within said length defined for said first RTP sub-extension; and
having subsequent RTP sub-extensions following the first RTP sub-extension.

9. A machine-readable medium that provides executable instructions, which when executed by a set of processors, cause said set of processors to perform negotiating operations for various types of streaming media data by a server comprising:

receiving a request for one or more types of streaming media data from a caching proxy server or a client, said request including a request for data associated with said streaming media data, said request including an identifier which represents one of several possible types

Google Exhibit No. 1008
Page 364

of data associated with said streaming media data;
determining if requested types of streaming media data are supported by the server; and
responding to the request with a response indicating the capability of the server to support the request.

10. A machine-readable medium that provides executable instructions, which when executed by a set of processors, cause said set of processors to perform negotiating operations for various types of streaming media data by a caching proxy server comprising:

sending a request for one or more types of related or unrelated streaming media data to a server, said request including a request for data associated with said streaming media data, said request including an identifier which represents one of several possible types of data associated with said streaming media data;
receiving a response to each requested type of streaming media data; and
deciding whether to proceed or terminate negotiation process associated with streaming media data.

11. A machine-readable medium that provides executable instructions, which when executed by a set of processors, cause said set of processors to perform frame thinning operations by a caching proxy server comprising:

receiving a message from a client to thin frames in a streaming media data transmission from said caching proxy server;
evaluating priority of frames; and
sending only selected frames.

12. A machine-readable medium that provides executable instructions, which when executed by a set of processors, cause said set of processors to send transmit time of RTP packet transmission operations from a caching proxy comprising:

requesting data corresponding to transmit time for streaming media data from a server;
receiving said streaming media data corresponding with transmit time information from the server;
storing the received information; and
transmitting from said caching proxy server to a client, said streaming media data at times specified by said transmit time.

13. A caching proxy server comprising:

means for sending a message for streaming media data to a server, said request including a request for data associated with said streaming media data, said request including an identifier which represents one of several possible types of data associated with said streaming media data;
means for receiving a response from the server indicating support for the requested streaming media data;
means for informing the server to send the supported data associated with said streaming media data;
means for receiving the streaming media data from the server;
means for receiving a request from the client to send streaming media data; and
means for sending the requested streaming media data to the client.

14. A RTP header comprising:

means for adding a first RTP sub-extension ID to an RTP header;
means for defining a length of said first RTP sub-extension by providing a sub-extension length;
means for providing data corresponding to the RTP sub-extension ID within said length defined for said first RTP sub-extension; and
means for having subsequent RTP sub-extensions following the first RTP sub-extension.

15. A server comprising:

means for receiving a request for one or more types of streaming media data from a caching proxy server or a client, said request including a request for data associated with said streaming media data, said request including an identifier which represents one of several possible types of data associated with said streaming media data;
means for determining if requested types of streaming media data are supported by the server; and
means for responding to the request with a response indicating the capability of the server to support the request.

16. A caching proxy server comprising:

means for sending a request for one or more types of related or unrelated streaming media data to a server, said request including a request for data associated with said streaming media data, said request including an identifier which represents one of several possible types

Google Exhibit No. 1008
Page 365

of data associated with said streaming media data;
means for receiving a response to each requested type of streaming media data; and
means for deciding whether to proceed or terminate negotiation process associated with streaming media data.

17. A frame thinning caching proxy server comprising:

means for receiving a message from a client to thin frames in a streaming media data transmission from said caching proxy server;
means for evaluating priority of frames; and
means for sending only selected frames.

18. A caching proxy server comprising:

means for requesting data corresponding to transmit time for streaming media data from a server;
means for receiving said streaming media data and corresponding transmit time information from the server;
means for storing the received information; and
means for transmitting from said caching proxy server to a client said streaming media data at times specified by said transmit time.

Google Exhibit No. 1008
Page 366

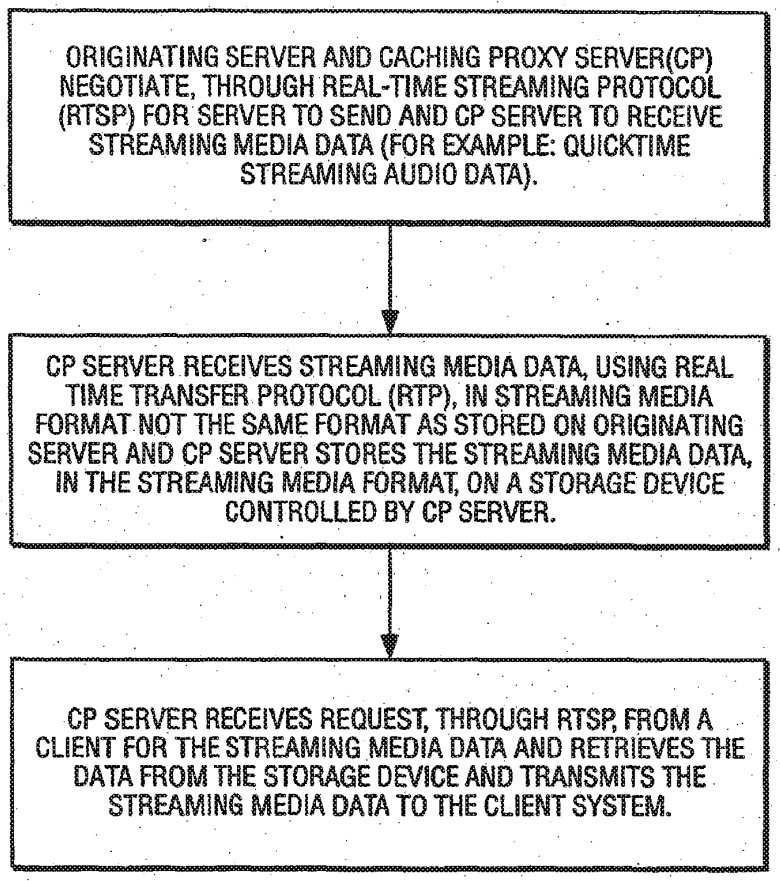


FIG. 1A

Google Exhibit No. 1008
Page 367

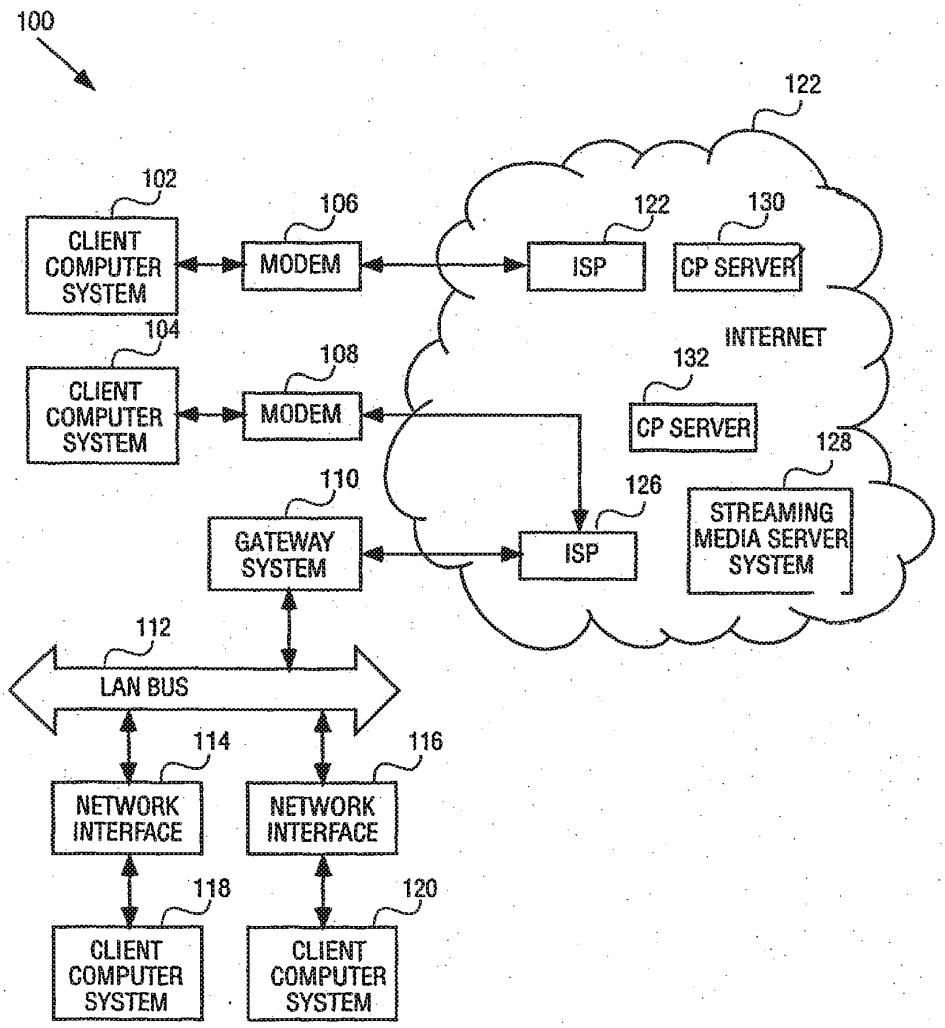


FIG. 1B

Google Exhibit No. 1008
Page 368

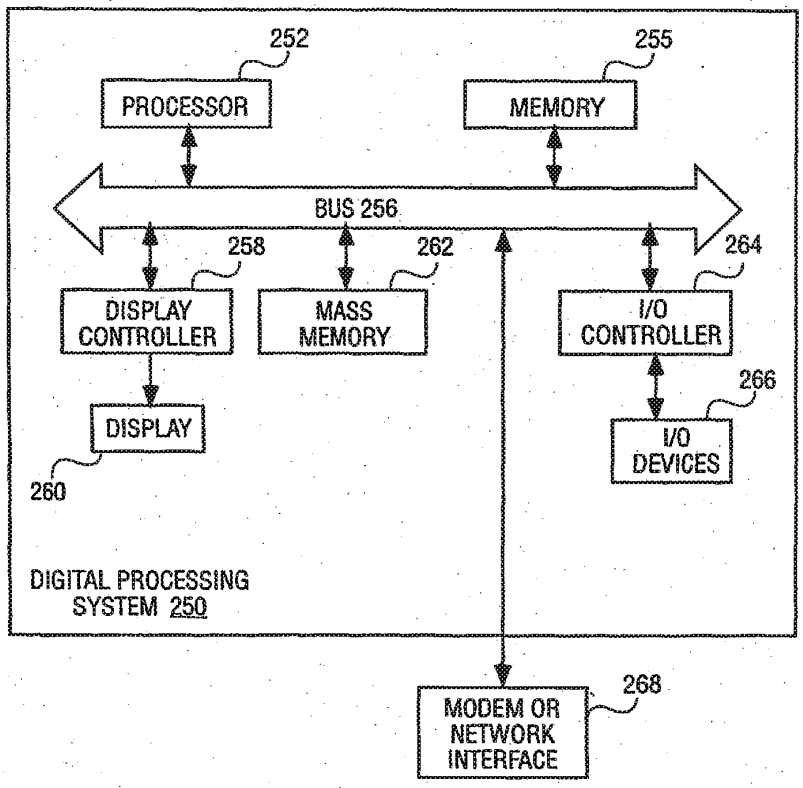


FIG. 2

Google Exhibit No. 1008
Page 369

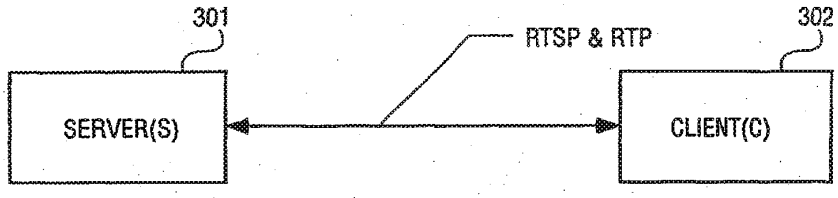


FIG. 3

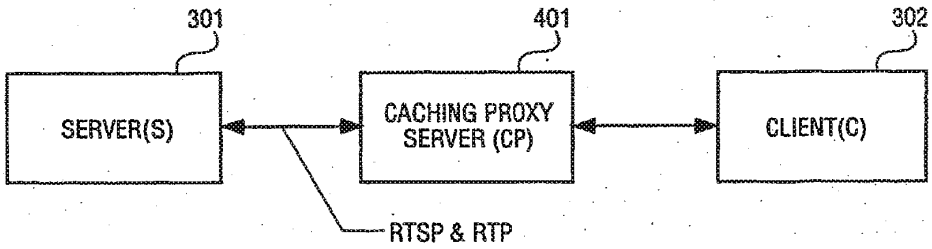


FIG. 4

Google Exhibit No. 1008
Page 370

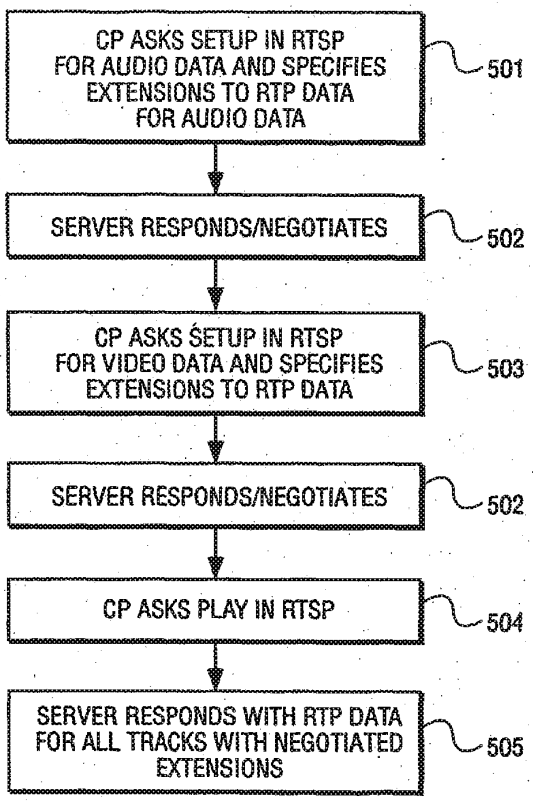


FIG. 5

Google Exhibit No. 1008
Page 371

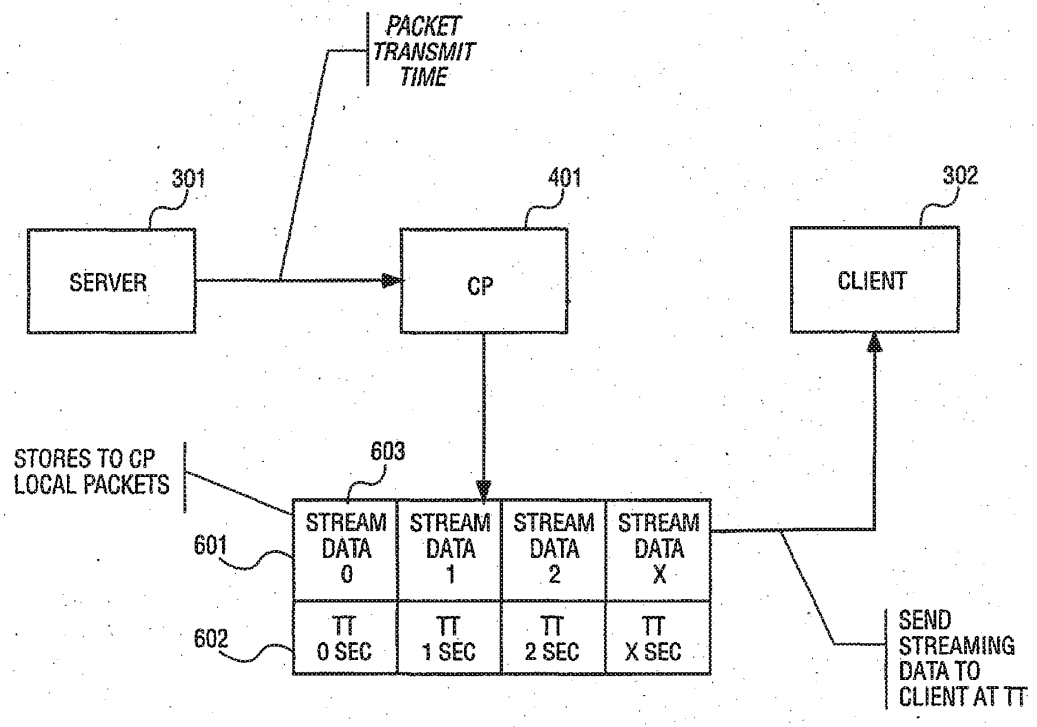

PACKET TRANSMIT TIME
301
401
302
SERVER
CP
CLIENT
STORES TO CP LOCAL PACKETS
603
601
602
STREAM DATA 0
STREAM DATA 1
STREAM DATA 2
STREAM DATA X
TT 0 SEC
TT 1 SEC
TT 2 SEC
TT X SEC
SEND STREAMING DATA TO CLIENT AT TT

FIG. 6

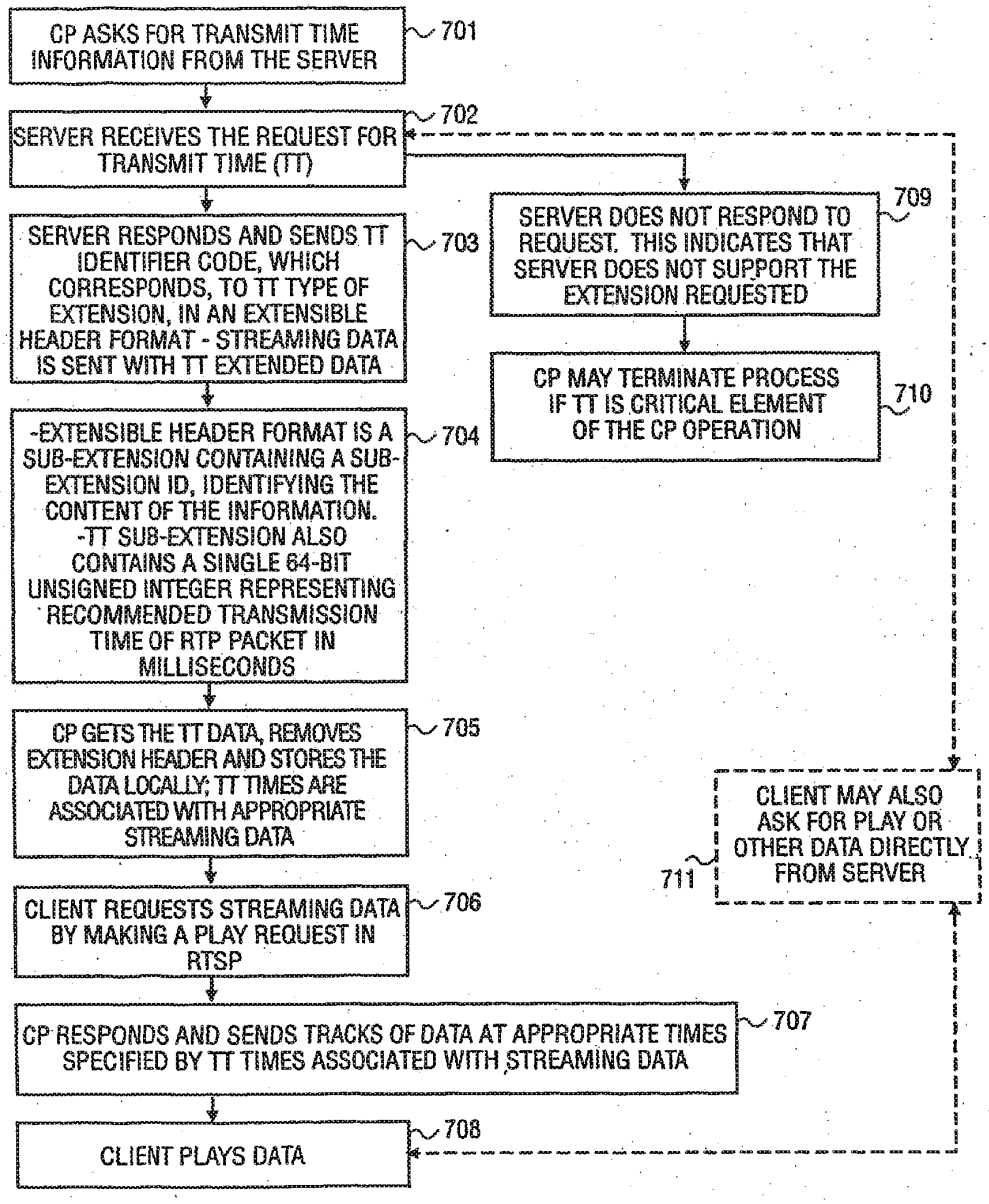


FIG. 7

Google Exhibit No. 1008
Page 373

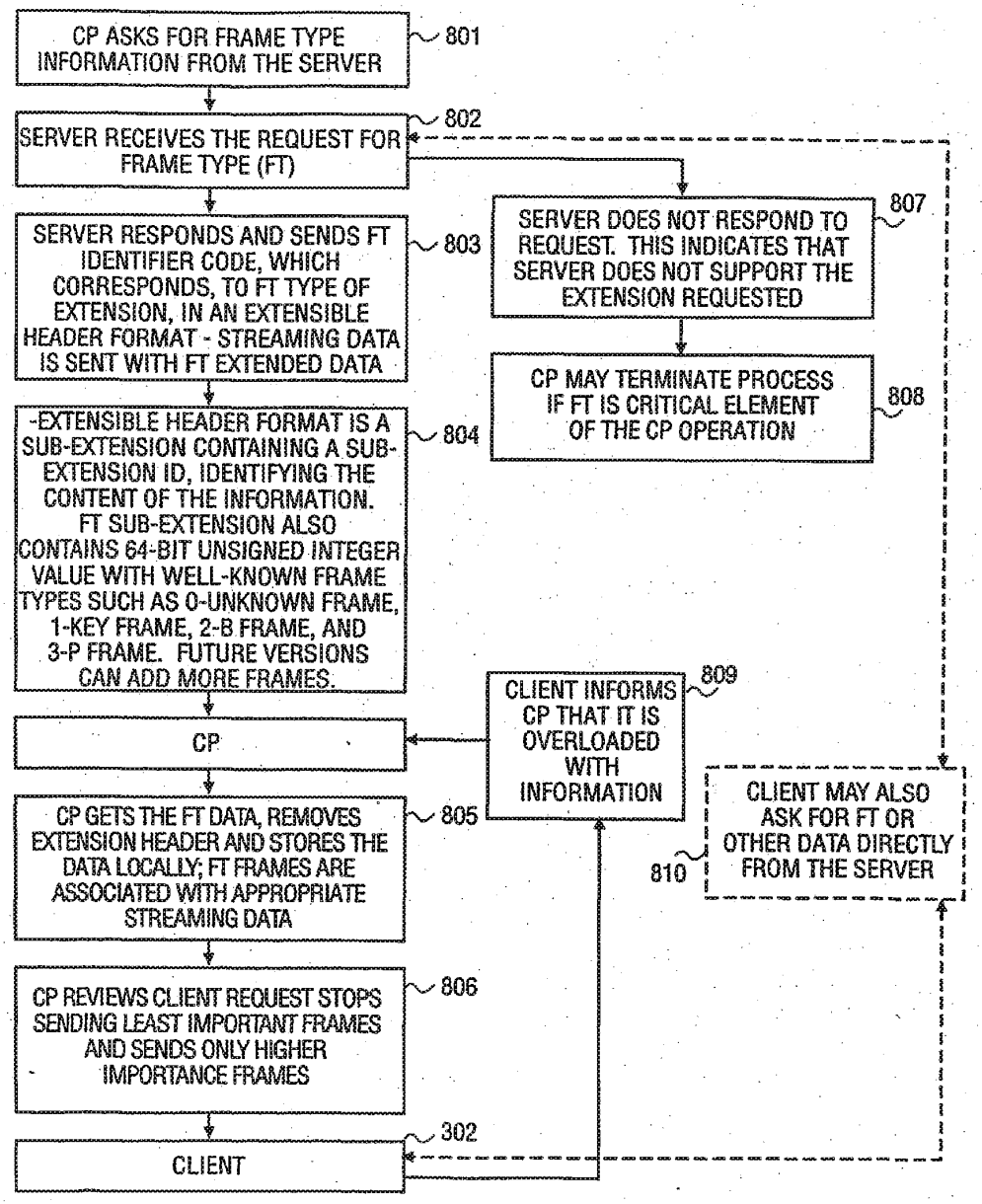


FIG. 8

Google Exhibit No. 1008
Page 374

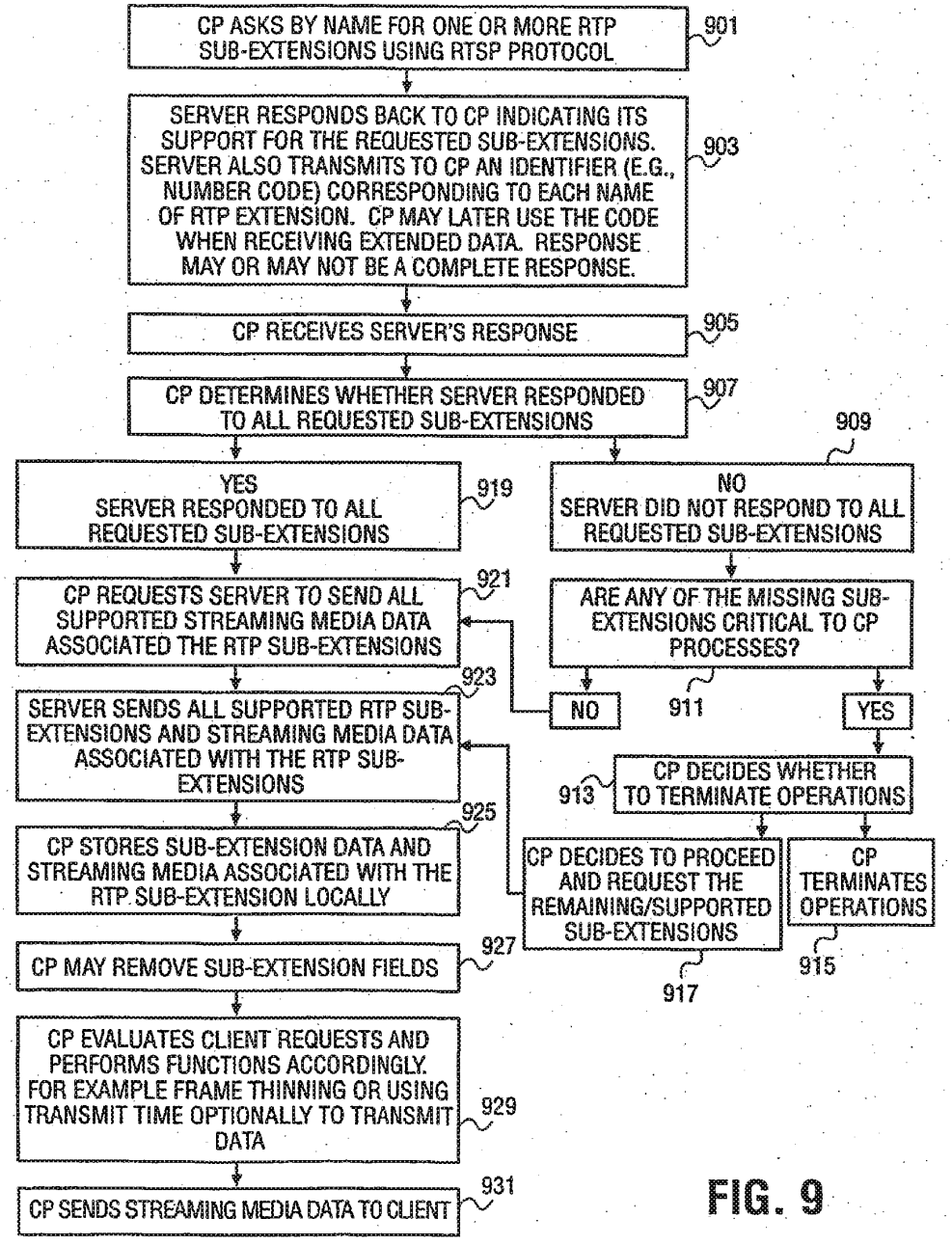


FIG. 9

Google Exhibit No. 1008
Page 375

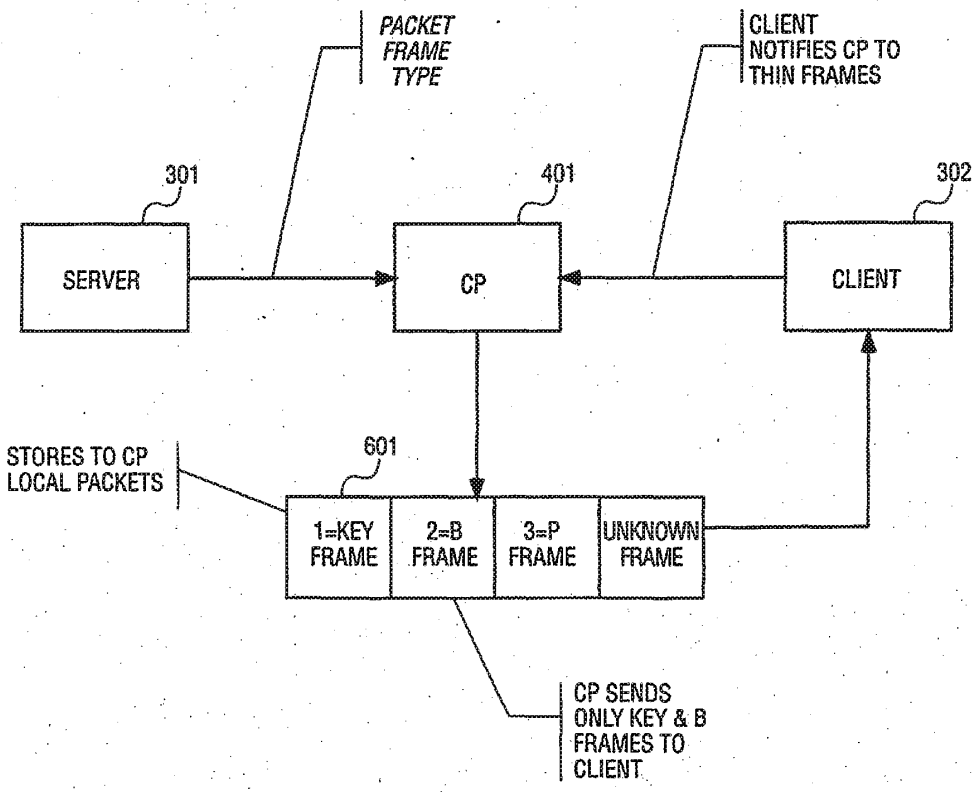


FIG. 10

Google Exhibit No. 1008
Page 376

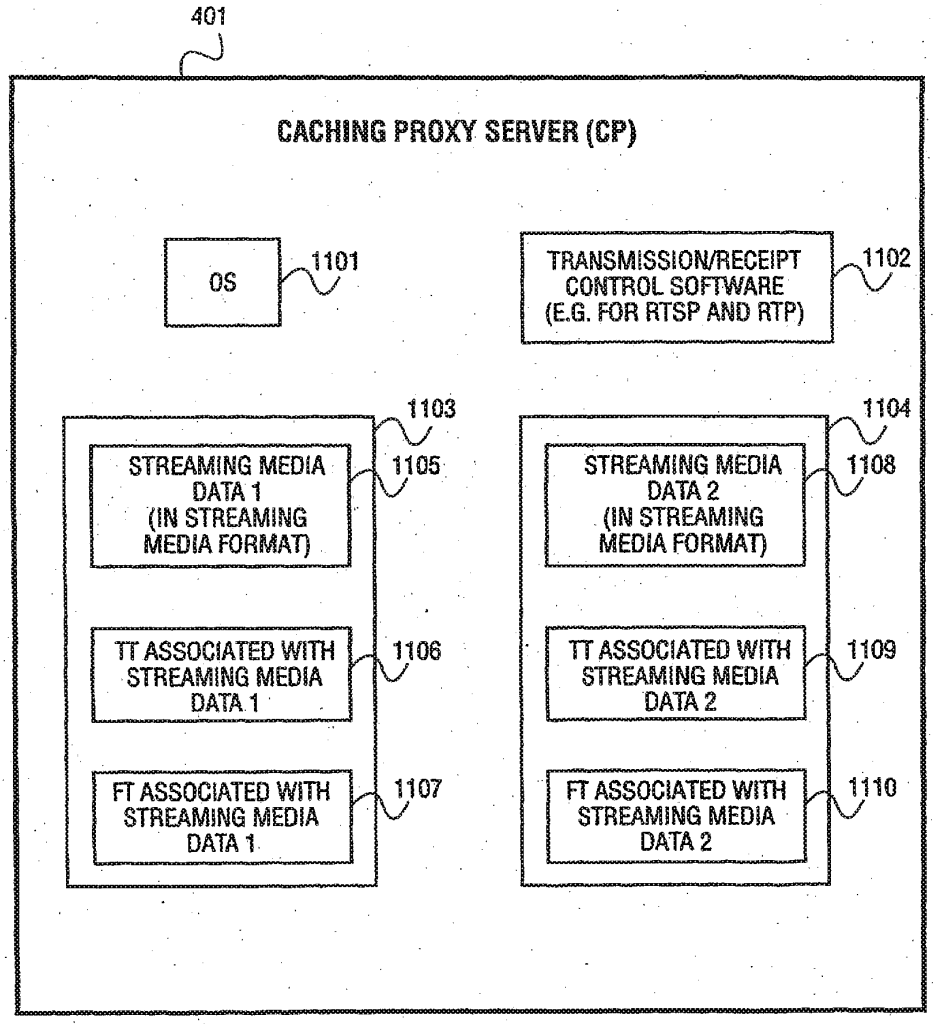


FIG. 11

Google Exhibit No. 1008
Page 377

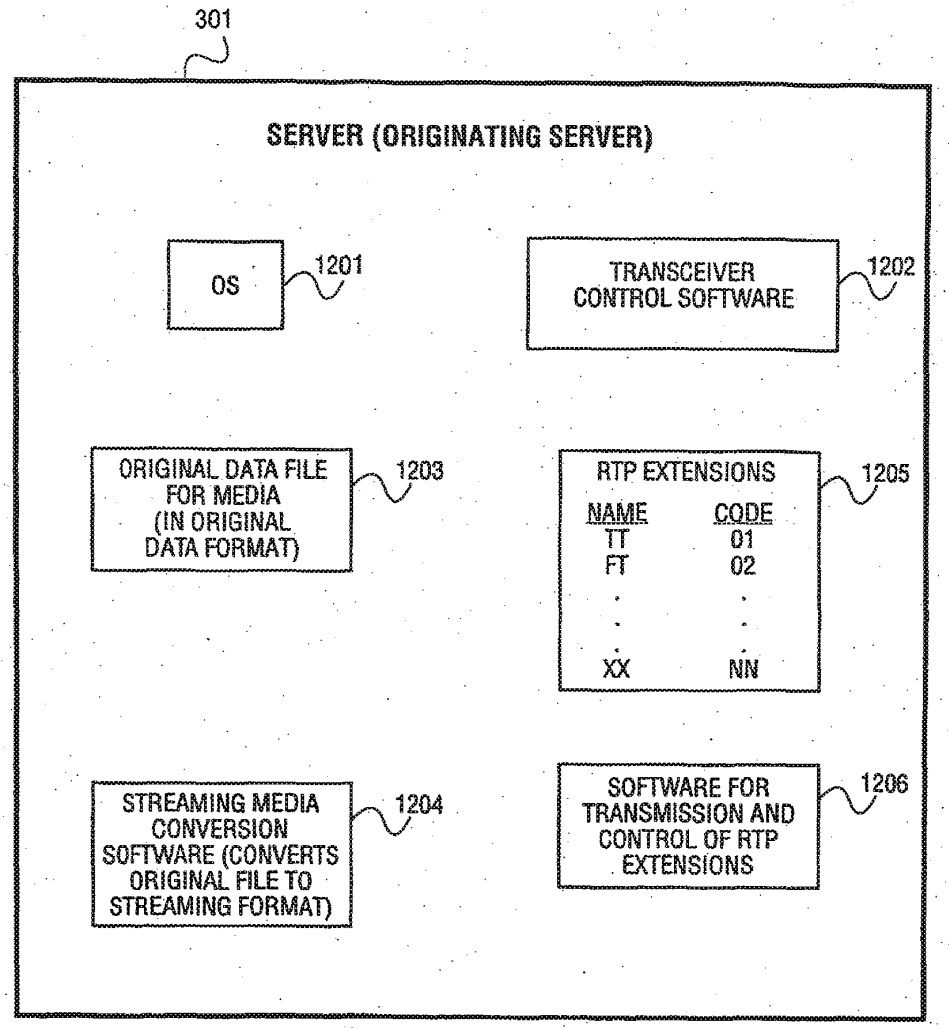


FIG. 12

Google Exhibit No. 1008
Page 378

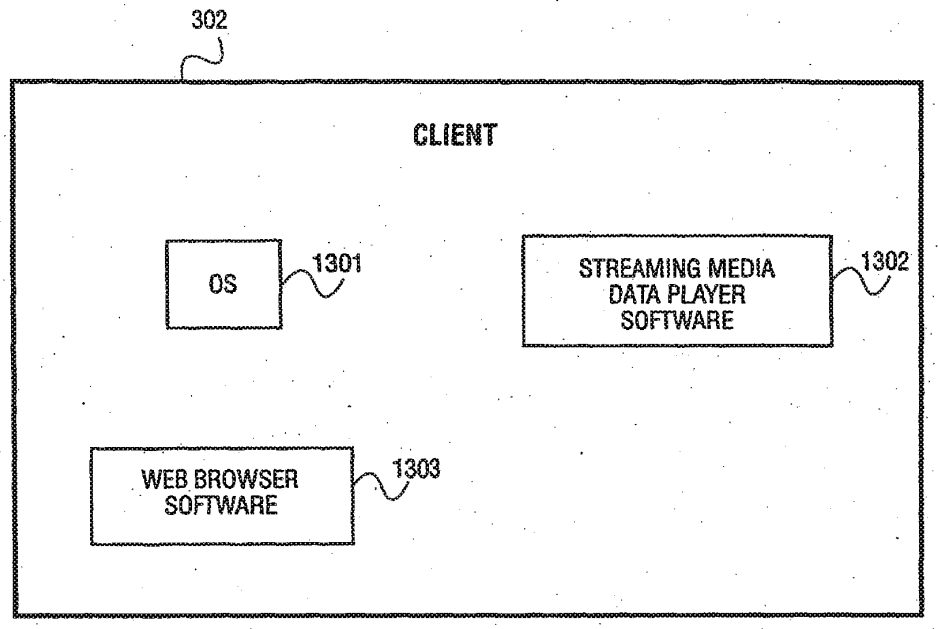


FIG. 13

Google Exhibit No. 1008
Page 379

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 03 02 7277

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
A	SCHULZRINNE H ET AL: "Internet Telephony: architecture and protocols - an IETF perspective" COMPUTER NETWORKS, ELSEVIER SCIENCE PUBLISHERS B.V., AMSTERDAM, NL, vol. 31, no. 3, 11 February 1999 (1999-02-11), pages 237-255, XP004304601 ISSN: 1389-1286 * paragraphs [0004] - [04.1] * * paragraph [05.1] * * paragraphs [05.4] - [05.5] * * paragraphs [0006], [0007] *	1-4, 6-10, 12-16,18	H04L29/06
A	US 5 918 020 A (HENTHORN WILLIAM JOHN ET AL) 29 June 1999 (1999-06-29) * column 6, lines 21-31 * * column 8, line 60 - column 9, line 16 * * column 11, line 42 - column 12, line 16 *	2-4, 8-10, 14-16	
A	WO 00/27087 A (DONNELL JOHN MARTIN O ;DALBY DAVID (GB); BRITISH TELECOMM (GB)) 11 May 2000 (2000-05-11) * page 5, line 30 - page 6, line 1 * * page 10, line 25 - page 11, line 3 *	2,8,14	TECHNICAL FIELDS SEARCHED (Int.Cl.7) H04L
A	SCHULZRINNE H ET AL: "RTP: A Transport Protocol for Real-Time Applications" RFC 1889: RTP, January 1996 (1996-01), XP002122581 USA * paragraphs [0005] - [5.3.1] *	2,8,14	

-/--

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
The Hague	12 February 2004	Dupuis, H

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

Google Exhibit No. 1008
Page 380

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 03 02 7277

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
A	WO 97/22201 A (XIE DONG ;CAMPBELL ROY H (US); CHEN ZHIGANG (US); TAN SEE MONG (US) 19 June 1997 (1997-06-19) * page 7, line 18 - page 9, line 10 * * page 17, line 20 - page 18, line 1 * * page 24, lines 6-8 * * page 25, line 17 * * page 27, lines 6-23 *	5,11,16, 17	

TECHNICAL FIELDS SEARCHED (Int.Cl.7)

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
The Hague	12 February 2004	Dupuis, H

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

Google Exhibit No. 1008
Page 381

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 03 02 7277

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

12-02-2004

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5918020 A	29-06-1999	NONE	
WO 0027087 A	11-05-2000	AU 6221099 A	22-05-2000
		CA 2347871 A1	11-05-2000
		EP 1125413 A1	22-08-2001
		WO 0027087 A1	11-05-2000
		JP 2002529966 T	10-09-2002
WO 9722201 A	19-06-1997	EP 0867003 A2	30-09-1998
		JP 2000515692 T	21-11-2000
		WO 9722201 A2	19-06-1997
		US 2003140159 A1	24-07-2003

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

Google Exhibit No. 1008
Page 382