Trials@uspto.gov
571-272-7822

Paper 42
Date: February 16, 2024

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SAMSUNG ELECTRONICS CO., LTD.,
Petitioner,

v.

MYPORT, INC.,
Patent Owner.

IPR2023-00023 (Patent 9,832,017 B2)
IPR2023-00024 (Patent 10,237,067 B2)
IPR2023-00025 (Patent 10,721,066 B2)[1]

Before JEAN R. HOMERE, TERRENCE W. McMILLIN, and SEAN P. O'HANLON, *Administrative Patent Judges*.

O'HANLON, *Administrative Patent Judge*.

TERMINATION
Due to Settlement After Institution of Trial
35 U.S.C. § 317; 37 C.F.R. § 42.74

[1] This Order applies to each of the listed proceedings. The parties are not authorized to use this caption for any subsequent papers absent prior authorization from the Board.

I. INTRODUCTION

On February 15, 2024, with Board authorization (*see* Paper 39), Petitioner and Patent Owner filed a joint motion to terminate the above-captioned proceedings.[2] Paper 40 ("Mot."). Along with the motion, the parties filed a copy of a Confidential Settlement Agreement (Ex. 1037), as well as a joint request to treat the Confidential Settlement Agreement as business confidential information (Paper 41). *See* 37 C.F.R. § 42.74(c) ("A party to a settlement may request that the settlement be treated as business confidential information and be kept separate from the files of an involved patent or application.").

II. DISCUSSION

Under 35 U.S.C. § 317(a), "[a]n inter partes review instituted under this chapter shall be terminated with respect to any petitioner upon the joint request of the petitioner and the patent owner, unless the Office has decided the merits of the proceeding before the request for termination is filed." It is also provided in 35 U.S.C. § 317(a) that if no petitioner remains, the Office may terminate the *inter partes* review.

In each of these proceedings, the Board issued a decision to institute *inter partes* review. Paper 21. However, the Board has not yet decided the merits of the proceedings or entered final written decisions. In the joint motion, the parties state that they have settled their dispute and have reached an agreement to request termination of these *inter partes* reviews. Paper 40,

[2] Citations herein refer to documents filed in IPR2023-00023. Similar documents were filed in the other proceedings.

1–2. Accordingly, in the circumstances present here, we determine that it is appropriate to terminate the proceedings. 37 C.F.R. § 42.71(a). This Order does not constitute a final written decision pursuant to 35 U.S.C. § 318(a).

The parties also ask that the Confidential Settlement Agreement be treated as business confidential information and be kept separate from the file of the challenged patent. Paper 41. We determine that it is appropriate to treat the Confidential Settlement Agreement as business confidential information, and to keep that document separate from the files of the challenged patent, pursuant to 37 C.F.R. § 42.74(c).

III. ORDER

Accordingly, it is:

ORDERED that Joint Motion to Terminate (Paper 40) is *granted*;

FURTHER ORDERED that the Joint Request to treat the Confidential Settlement Agreement as business confidential information (Paper 41) is *granted*;

FURTHER ORDERED that the Confidential Settlement Agreement (Ex. 1037) shall remain designated as "Parties and Board Only" in the Board's electronic filing system, be kept separate from the files of the involved patent, and be made available only under the provisions of 35 U.S.C. § 317(b) and 37 C.F.R. § 42.74(c); and

FURTHER ORDERED that IPR2023-00023, IPR2023-00024, and IPR2023-00025 are *terminated*.

IPR2023-00023 (Patent 9,832,017 B2)
IPR2023-00024 (Patent 10,237,067 B2)
IPR2023-00025 (Patent 10,721,066 B2)

For PETITIONER:

Andrew R. Sommer
Jay Guiliano
Harry Park
Philip Hoffmann
Jenna Kuh
GREENBERG TRAURIG, LLP
sommera@gtlaw.com
jay.guiliano@gtlaw.com
parkh@gtlaw.com
hoffmannp@gtlaw.com
kuhj@gtlaw.com

For PATENT OWNER:

Matthew Phillips
Kevin Laurence
Derek Meeker
LAURENCE & PHILLIPS IP LAW
mphillips@lpiplaw.com
klaurence@lpiplaw.com
dmeeker@lpiplaw.com

John Lord
Sarah E. Spires
SKIERMONT DERBY LLP
jlord@skiermontderby.com
sspires@skiermontderby.com

Disclaimer

9,832,017 B2 - Michael F. Malone, McKinney, TX (US). APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSATION PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/CONTEXTUAL/TAG(S), STORAGE AND SEARCH RETRIEVAL. Patent dated November 28, 2017. Disclaimer filed January 24, 2023, by the assignee, MyPort, Inc.

I hereby disclaim the following complete claims 1-5 of said patent.

(Official Gazette, May 16, 2023)

Disclaimer

9,832,017 B2 - Michael F. Malone, McKinney, TX (US). APPRATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTOO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL. Patent dated November 28, 2017. Disclaimer filed January 24, 2023, by the inventor.

I hereby disclaim the following complete claims 1-5, of said patent.

(Official Gazette, May 2, 2023)

Trials@uspto.gov
571-272-7822

Paper 21
Date: April 24, 2023

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SAMSUNG ELECTRONICS CO., LTD.,
Petitioner,

v.

MYPORT, INC.,
Patent Owner.

IPR2023-00023
Patent 9,832,017 B2

Before JEAN R. HOMERE, TERRENCE W. McMILLIN, and SEAN P. O'HANLON, *Administrative Patent Judges*.

O'HANLON, *Administrative Patent Judge*.

DECISION
Granting Institution of *Inter Partes* Review
35 U.S.C. § 314

I. INTRODUCTION

A. Background

Samsung Electronics Co., Ltd. ("Petitioner") filed a Petition for *inter partes* review of claims 1–17 of U.S. Patent No. 9,832,017 B2 (Ex. 1001, "the '017 patent"). Paper 1 ("Pet."), 1. MyPort, Inc. ("Patent Owner") filed a Preliminary Response. Paper 12 ("Prelim. Resp."). With our authorization (*see* Ex. 3002), Petitioner filed a Preliminary Reply (Paper 18, "Prelim. Reply") and Patent Owner filed a Preliminary Sur-reply (Paper 19, "Prelim. Sur-reply").

Institution of an *inter partes* review is authorized by statute only when "the information presented in the petition . . . and any response . . . shows that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition." 35 U.S.C. § 314(a) (2018). We have authority, acting on the designation of the Director, to determine whether to institute an *inter partes* review under 35 U.S.C. § 314 and 37 C.F.R. § 42.4(a). For the reasons set forth below, upon considering the parties' briefs and evidence of record, we conclude that the information presented shows that there is a reasonable likelihood that Petitioner would prevail in establishing the unpatentability of at least one of the challenged claims. Thus, we institute *inter partes* review.

B. Real Parties in Interest

Petitioner identifies itself and Samsung Electronics America, Inc. as real parties in interest. Pet. 63.

Patent Owner identifies itself as the sole real party in interest. Paper 4, 2.

C. Related Matters

The parties indicate that the '017 patent is the subject of the following district court proceeding:

> *MyPort, Inc. v. Samsung Electronics Co., Ltd.*, Case No. 2:22-cv-114 (E.D. Tex. filed April 15, 2022) ("the Texas litigation").

Pet. 63; Paper 4, 2. The parties note other petitions for *inter partes* review filed by Petitioner and challenging patents owned by Patent Owner. Pet. 63; Paper 4, 2. More particularly, Petitioner filed a petition for *inter partes* review (IPR2023-00024) of claims 1–17 of U.S. Patent No. 10,237,067 and a petition for *inter partes* review (IPR2023-00025) of claims 1–17 of U.S. Patent No. 10,721,066. Pet. 63; Paper 4, 2.

D. The Challenged Patent

The '017 patent relates to the storage and retrieval of digital media files. Ex. 1001, 2:15–18. Digital media devices, such as mobile phones with audio and video recording capability, have greatly reduced the cost of creating new image, movie, and audio recordings, making the number of these recordings available in any given individual's library skyrocket. *Id.* at 2:22–31. This ability to store virtually unlimited numbers of media files, however, has created new problems. *Id.* at 2:32–33. For example, having a large number of media files increases the difficulty of manually describing and indexing the files. *Id.* at 2:33–35. Thus, "for many, photographs, movies and audio recordings are kept in poorly organized computer files and finding any particular picture, movie or recording is a difficult and time-consuming process." *Id.* at 2:34–39.

The '017 patent purports to provide an improved system having a capture device that includes a data converter for capturing various types of information, such as "[s]till pictures, moving pictures, audio, [and] telemetry," and organizing the information into a data element. Ex. 1001. 3:52–59. The data converter can be a digital camera, digital video recorder, or "any type of device that will capture the information and place it in some type of digitized format." *Id.* at 3:60–65. Metadata, such as time and date, location, operator identification, or other information relevant to establish the context of the data element, is created at the same time as the data element. *Id.* at 4:4–9. The capture device may also include a secondary data converter for gathering additional information that relates to defining the context of the data element. *Id.* at 4:24–27. "For example, after a camera/video recorder takes a picture/video, a microphone (the secondary data converter) might capture the audio describing the image or the audio from the video just captured." *Id.* at 4:27–31. This additional information is stored as a "context description element" with the data element and metadata. *Id.* at 4:31–32.

The capture device may send this stored data to a storage facility that may add "context tags," which are searchable elements derived from the different types of data. Ex. 1001, 4:66–5:3, 5:36–41. As an example, for a data element that "is a still photograph or video, the storage facility may create context tags that describe elements of the scene or image(s), such as 'animal,' or 'dog,' or 'Spot,' depending on the mechanism that converts the information in the data element or the context description element into a tag." *Id.* at 5:41–46. Context tags may be created using "a speech-to-text algorithm" for audio files or "image recognition and identification" for still

or moving images. *Id.* at 5:47–51. The result is a set of data for storage that "now includes a set of searchable tags specific to that image, video, audio or other media." *Id.* at 5:51–55.

Figure 2 illustrates "a diagrammatic view of the overall operation of the system," and is reproduced below. Ex. 1001, 6:32–33.

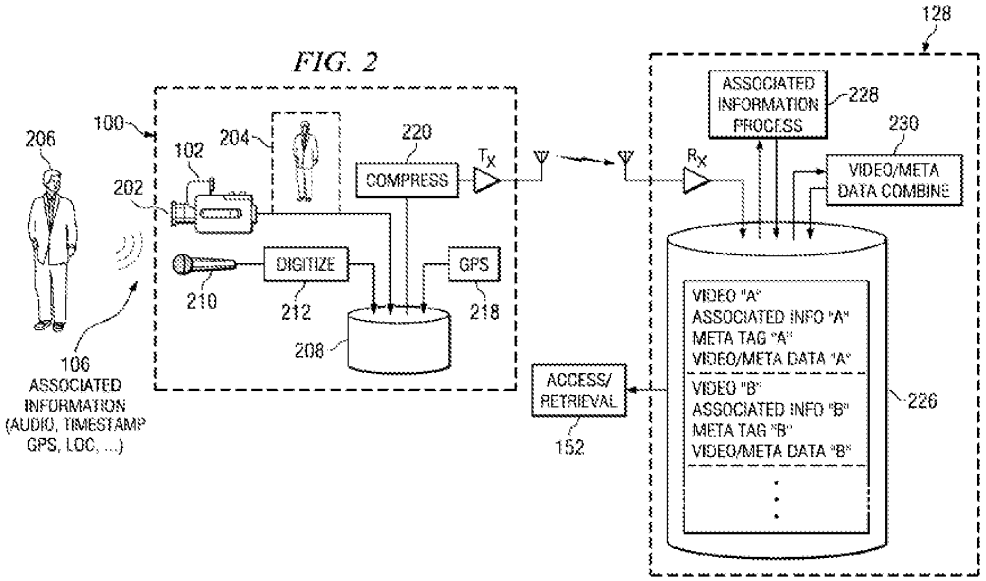


Figure 2 shows capture device system 100 including capture device 102, illustrated as a digital camera, and microphone 210. *Id.* at 6:32–48. All the information collected via the capture device and microphone is stored in database 208 within the capture device, from which location the information may be compressed and then sent to storage facility 128 for storage in database 226. *Id.* at 6:57–7:26. "At this time there will be, for example, a video clip or a video image (such as a still image) stored in association with the various information that is associated therewith." *Id.* at 7:26–29. Access and retrieval block 152 "can allow one to access the particular modified or 'tagged' image via a search," so that "[a]ny information that can be provided

over and above the video information that was provided in all of the associated information at the camera can then be searched." *Id.* at 7:51–60. In an alternative embodiment, illustrated in Figure 3, databases 208 and 226 are combined into a single database within the capture device, and the capture device creates and adds context tags to the image or video. *Id.* at 7:62–8:17; Fig. 3.

E. Disclaimer of Claims 1–5

A "patent owner may file a statutory disclaimer under 35 U.S.C. [§] 253(a) in compliance with § 1.321(a) of this chapter, disclaiming one or more claims in the patent. No *inter partes* review will be instituted based on disclaimed claims." 37 C.F.R. § 42.107(e). A disclaimer under 35 U.S.C. § 253(a) is "considered as part of the original patent" as of the date on which it is "recorded" in the U.S. Patent and Trademark Office ("Office"). 35 U.S.C. § 253(a). For a disclaimer to be "recorded" in the Office, the document filed by the patent owner must:

> (1) Be signed by the patentee, or an attorney or agent of record;
>
> (2) Identify the patent and complete claim or claims, or term being disclaimed. A disclaimer which is not a disclaimer of a complete claim or claims, or term will be refused recordation;
>
> (3) State the present extent of patentee's ownership interest in the patent; and
>
> (4) Be accompanied by the fee set forth in [37 C.F.R.] § 1.20(d).

37 C.F.R. § 1.321(a); *see also Vectra Fitness, Inc. v. TNWK Corp.*, 162 F.3d 1379, 1382 (Fed. Cir. 1998) (holding that a § 253 disclaimer is "recorded"

on the date that the Office receives a disclaimer meeting the requirements of 37 C.F.R. § 1.321(a), and that no further action is required in the Office for a disclaimer to be "recorded").

Here, Patent Owner filed a statutory disclaimer of claims 1–5 of the '017 patent. Prelim. Resp. 8–9; Ex. 2001. Based on our review of Exhibit 2001, we determine that a disclaimer of claims 1–5 of the '017 patent under 35 U.S.C. § 253(a) has been filed with the Office as of January 24, 2023. Based on the information in the public record, we find that the disclaimer complies with the above-listed requirements of 37 C.F.R. § 1.321(a).

Because claims 1–5 have been disclaimed under 35 U.S.C. § 253(a), in compliance with 37 C.F.R. § 1.321(a), we do not reach challenges to claims 1–5. As a result, the remaining challenged claims are claims 6–17 ("the challenged claims").

F. The Challenged Claims

Petitioner challenges claims 6–17 of the '017 patent. Pet. 1. Claims 6 and 13 are independent. Claim 6 is illustrative of the challenged claims and is reproduced below.

> 6. A system for capturing image and audio information for storage, comprising:
>
> a capture device having:
>
> internal storage;
>
> a microphone interfacable with [an] external audio information source that generates external audio information and a first data converter for capturing the first external audio information from the microphone,

> a camera interfacing with [an] external image source to capture an image therefrom;
>
> the first data converter processing the captured external audio information and storing it in a first digital audio format as stored digital audio in internal storage within the capture device, the camera for processing the captured image and storing it as a stored digital image in internal storage and a combiner for generating an association between the stored digital audio and the stored digital image,
>
> a media data converter for converting the received set of captured information to convert the received digital audio to a text based searchable file as a text context tag and creating an image recognition searchable context tag with image recognition of at least a portion of the digital image and associating the text and image recognition context tags with the digital image, and
>
> the internal storage storing the digital image in association with the text and image recognition context tags.

Ex. 1001, 11:6–35.

G. Asserted Grounds of Unpatentability

The Petition relies on the following prior art references:

Name	Reference	Exhibit
Fuller	US 6,833,865 B1, issued December 21, 2004	1003
Rothschild	US 2006/0114338 A1, published June 1, 2006	1007
Anderson	US 6,499,016 B1, issued December 24, 2002	1008
Camahan	US 2005/0267894 A1, published December 1, 2005	1010
Davis	US 2002/0001395 A1, published January 3, 2002	1013

Petitioner asserts the following grounds of unpatentability:[1]

Claim(s) Challenged	35 U.S.C. §	Reference(s)
6–17	103(a)[2]	Rothschild, Anderson
6–9, 13–17	102(b)	Fuller
6–9, 13–17	103(a)	Fuller, Camahan
10–12	103(a)	Fuller, Davis
10–12	103(a)	Fuller, Camahan, Davis

Pet. 1. Petitioner submits a declaration of Benjamin B. Bederson, Ph.D. (Ex. 1002, "Bederson Declaration") in support of its contentions.

II. DISCRETIONARY DENIAL OF INSTITUTION

Under 35 U.S.C. § 314(a), the Director has discretion to deny institution. In determining whether to exercise that discretion on behalf of the Director, we are guided by the Board's precedential decision in *NHK Spring Co., Ltd. v. Intri-Plex Technologies, Inc.*, IPR2018-00752, Paper 8 (PTAB Sept. 12, 2018) (designated precedential).

In *NHK*, the Board found that the "advanced state of the district court proceeding" was a "factor that weighs in favor of denying" the petition under § 314(a). *NHK*, Paper 8 at 20. The Board determined that "[i]nstitution of an *inter partes* review under these circumstances would not be consistent with 'an objective of the AIA . . . to provide an effective and

[1] A challenge to claims that Patent Owner has disclaimed (claims 1–5) is not listed here.

[2] The application resulting in the '017 patent claims priority to a date prior to the date when the Leahy-Smith America Invents Act ("AIA"), Pub. L. No. 112–29, 125 Stat. 284 (2011), took effect. Thus, we refer to the pre-AIA version of sections 102 and 103.

efficient alternative to district court litigation.'" *Id.* (citing *Gen. Plastic Indus. Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19 at 16–17 (designated precedential in relevant part)).

"[T]he Board's cases addressing earlier trial dates as a basis for denial under *NHK* have sought to balance considerations such as system efficiency, fairness, and patent quality." *Apple Inc. v. Fintiv Inc.*, IPR2020-00019, Paper 11 at 5 (PTAB Mar. 20, 2020) (designated precedential). *Fintiv* sets forth six non-exclusive factors for determining "whether efficiency, fairness, and the merits support the exercise of authority to deny institution in view of an earlier trial date in the parallel proceeding." *Id.* at 6. These factors are reproduced below:

> 1. whether the court granted a stay or evidence exists that one may be granted if a proceeding is instituted;
>
> 2. proximity of the court's trial date to the Board's projected statutory deadline for a final written decision;
>
> 3. investment in the parallel proceeding by the court and the parties;
>
> 4. overlap between issues raised in the petition and in the parallel proceeding;
>
> 5. whether the petitioner and the defendant in the parallel proceeding are the same party; and
>
> 6. other circumstances that impact the Board's exercise of discretion, including the merits.

We discuss the parties' arguments in the context of considering the above factors. In evaluating the factors, we take a holistic view of whether efficiency and integrity of the system are best served by denying or instituting review. *Fintiv*, Paper 11 at 6. We are also guided by the Director's Memorandum, *Interim Procedure for Discretionary Denials in*

AIA Post-Grant Proceedings with Parallel District Court Litigation, issued June 21, 2022 ("Guidance Memo"),[3] which provides several clarifications concerning the application of the *Fintiv* factors.

A. Whether the Court Granted a Stay or Evidence Exists that One May Be Granted if a Proceeding Is Instituted

The existence of a district court stay pending Board resolution of an *inter partes* review has weighed strongly against discretionary denial, while a denial of such a stay request sometimes weighs in favor of discretionary denial. *Fintiv*, Paper 11 at 6–8.

Patent Owner argues that this factor favors denial of institution because Petitioner has presented no evidence regarding a stay. Prelim. Resp. 52. According to Patent Owner, "a stay in [the Texas litigation] is extremely unlikely, as the [P]etitioner tacitly acknowledged when it chose not to file a reply brief in support of its stay motion." *Id.*

Petitioner replies that its "motion to stay remains pending." Prelim. Reply 4.

Patent Owner replies that Petitioner has failed to produce evidence that its motion to stay the Texas litigation will be granted if we institute review in this proceeding. Prelim. Sur-reply 1.

The record indicates that Petitioner has filed a motion to stay the Texas litigation (*see* Ex. 2011) and Patent Owner has filed an opposition to Petitioner's motion (*see* Ex. 2012). The likelihood of the district court

[3] *Available at* https://www.uspto.gov/sites/default/files/documents/interim_proc_discretionary_denials_aia_parallel_district_court_litigation_memo_20220621_.pdf.

granting Petitioner's motion is speculative. Accordingly, the facts underlying this factor are neutral.

B. Proximity of the Court's Trial Date to the Board's Projected Statutory Deadline for a Final Written Decision

When a district court's trial date will occur before the projected statutory deadline for the final written decision, the Board generally weighs this factor in favor of exercising discretion to deny institution. *Fintiv*, Paper 11 at 9. "If the court's trial date is at or around the same time as the projected statutory deadline or even significantly after the projected statutory deadline, the decision whether to institute will likely implicate other factors discussed herein, such as the resources that have been invested in the parallel proceeding." *Id.* When considering the facts underlying this factor, we "will consider the median time from filing to disposition of the civil trial for the district in which the parallel litigation resides." Guidance Memo 3. "[W]hen other relevant factors weigh against exercising discretion to deny institution or are neutral, the proximity to trial should not alone outweigh all of those other factors." *Id.* at 8. "Parties may present evidence regarding the most recent statistics on median time-to-trial for civil actions in the district court in which the parallel litigation resides for [our] consideration." *Id.* at 8–9 (footnote omitted).

Petitioner acknowledges that "trial [in the Texas litigation] is . . . about six weeks before an anticipated final written decision," but argues that "trial dates are unpredictable, and the median time to trial in the Eastern District of Texas is 24.5 months, placing trial more than a month after a [final written decision], thus favoring institution." Pet. 62–63 (citing Ex. 1023; Guidance Memo 3, 8–9).

Patent Owner argues that this factor favors denial because "the [Texas litigation] is scheduled to begin two months and three weeks *before* the final written decision would be due in this case." Prelim. Resp. 52 (citing *id.* at 11; Ex. 2013, 1). Patent Owner argues that Petitioner's calculation of the projected trial date is incorrect and asserts that the median time of 24.5 months puts the trial date "just two business days after the [final written decision] deadline." *Id.* at 53. Continuing, Patent Owner argues that "the median time to trial has been skewed recently because of the coronavirus pandemic," and that, prior to the pandemic, "the median time to trial in the U.S. District Court for the Eastern District of Texas was only 17.5 months." *Id.* (citing Ex. 1023, 35).

Petitioner replies that "the Director's [Guidance] Memo accounted for [delays due to the coronavirus pandemic] when stating that the Office would rely on median times to trial." Prelim. Reply 5.

Patent Owner replies that "nothing in the [Guidance] Memo condones reliance on skewed statistics." Prelim. Sur-reply 1.

The latest statutory due date for a final written decision in this proceeding is April 26, 2024. The Texas litigation is scheduled to begin on February 5, 2024. Ex. 2013, 1. However, Petitioner presents evidence that, as of June 30, 2022, the median time-to-trial for civil actions in the Eastern District of Texas is 24.5 months (Ex. 1023, 35), which, as Patent Owner concedes, would place the trial date two days after the latest statutory due date for a final written decision in this proceeding (Prelim. Resp. 53). The most recent statistics show that, as of December 31, 2022, the median time-to-trial date for civil actions in the Eastern District of Texas is 19.0 months

(Ex. 3003, 35), which would place the trial date prior to the February 5, 2024, scheduled date.

Accordingly, the facts underlying this factor weigh in favor of exercising our discretion to deny institution. However, the Guidance Memo instructs us that, "when analyzing the proximity of the court's trial date under factor two of *Fintiv*, when other relevant factors weigh against exercising discretion to deny institution or are neutral, the proximity to trial should not alone outweigh all of those other factors." Guidance Memo 8.

C. Investment in the Parallel Proceeding by the Court and the Parties

If, at the time of the institution decision, the district court has issued substantive orders related to the challenged patent, such as a claim construction order, this fact weighs in favor of denial. *See Fintiv*, Paper 11 at 9–10. On the other hand, if the district court has not issued such orders, this fact weighs against discretionary denial. *Id.* at 10. Additionally, we also consider Petitioner's diligence in filing the Petition in weighing this factor. *Fintiv*, Paper 11 at 11 ("If the evidence shows that the petitioner filed the petition expeditiously, such as promptly after becoming aware of the claims being asserted, this fact has weighed against exercising the authority to deny institution under *NHK*.").

Petitioner argues that "the parties and court have not significantly invested in the case because discovery is, at best, in its infancy, invalidity contentions are not due for a month, and this Petition is being filed within a

month of receiving infringement contentions, thus favoring institution." Pet. 63 (citing Ex. 1024, 6).[4]

Patent Owner argues that, "[b]y the time the decision on institution is due in this case, the parties and the court will have invested significant time and energy in the case including, for example, exchange of infringement and invalidity contentions, engaging in fact discovery, and beginning the claim construction process." Prelim. Resp. 53–54.

Petitioner replies that Patent Owner "has not identified any efforts it has made to litigate the validity issue, which is the relevant question under *Fintiv* factor 3." Prelim. Reply 5 (citing *Sand Revolution II, LLC v. Cont'l Intermodal Grp.–Trucking LLC*, IPR2019-01393, Paper 24 at 11–12 (PTAB June 16, 2020) (designated informative)).

Patent Owner replies that, "[b]y the time the decision on institution is made, the parties will have served invalidity contentions; engaged in fact discovery regarding, *inter alia*, validity; and begun the claim construction process, which affects validity and overlaps with the IPR." Prelim. Sur-reply 1.

In the Texas litigation, the parties have exchanged infringement and invalidity contentions. Ex. 2013, 4; Ex. 2004. Claim construction does not appear to be an issue, as the parties have not sought construction of any term in this proceeding. *See generally* Prelim. Resp.; Pet. 2. Fact and expert discovery, expert reports, dispositive motions, and, if applicable, claim construction briefing will continue well after the Institution Decision

[4] We note that page six of Exhibit 1024 is the signature page of Patent Owner's disclosures in the Texas litigation and, thus, does not appear to support Petitioner's assertions.

deadline. Ex. 2013, 2–5. Thus, the parties have engaged in some work in the Texas litigation, but significant work still remains to be done.

Patent Owner filed its complaint in the Texas litigation on April 15, 2022. Ex. 1015, 35. Petitioner diligently filed its Petition on October 12, 2022 (Paper 3, 1), approximately one month after Patent Owner filed its infringement contentions (*see* Ex. 2013, 4) and six months prior to the statutory deadline for filing the Petition (*see* 35 U.S.C. § 315(b)).

Accordingly, the facts underlying this factor weigh against exercising our discretion to deny institution.

D. Overlap Between Issues Raised in the Petition and in the Parallel Proceeding

"[I]f the petition includes the same or substantially the same claims, grounds, arguments, and evidence as presented in the parallel proceeding, this fact has favored denial." *Fintiv*, Paper 11 at 12. "Conversely, if the petition includes materially different grounds, arguments, and/or evidence than those presented in the district court, this fact has tended to weigh against exercising discretion to deny institution under *NHK*." *Id.* at 12–13.

Concerns about the degree of overlap may be mitigated where a petitioner agrees not to pursue in the parallel district court litigation the grounds advanced in the petition. *Sand Revolution*, IPR2019-01393, Paper 24 at 11–12. The Board suggested that a broader stipulation such as one stipulating that the petitioner "would not pursue any ground raised or that could have been reasonably raised in an IPR, i.e., any ground that could be raised under §§ 102 or 103 on the basis of prior art patents or printed publications" would better address concerns relating to overlapping issues.

Id. at 12 n.5; *see also* Guidance Memo 7 (explaining that such a broad stipulation obviates a full analysis of the *Fintiv* factors).

Petitioner concedes that the Petition includes the same or substantially the same claims, grounds, arguments, and evidence as it presents in the Texas litigation. Prelim. Reply 5 (asserting that Petitioner "will withdraw its contentions *based on the grounds it has presented in the Petition* in the [Texas] litigation" (emphasis added)). However, Petitioner's stipulation reduces the overlap relating to the challenges presented in the Petition and mitigates "to some degree the concerns of duplicative efforts between the district court and the Board, as well as concerns of potentially conflicting decisions." *See Sand Revolution*, IPR2019-01393, Paper 24 at 12.

Accordingly, the facts underlying this factor weigh marginally against exercising our discretion to deny institution.

E. Whether the Petitioner and the Defendant in the Parallel Proceeding Are the Same Party

"If a petitioner is unrelated to a defendant in an earlier court proceeding," this fact has weighed against exercising discretion to deny institution. *Fintiv*, Paper 11 at 13–14.

There is no dispute that the parties in this proceeding are identical to the parties in the Texas litigation. *See* Prelim. Resp. 58; Pet. 63. Thus, the facts underlying this factor are neutral.

F. Merits and Other Circumstances that Impact the Board's Exercise of Discretion

Petitioner argues that this factor favors institution because its "Petition presents a strong case of unpatentability." Pet. 63.

Patent Owner argues that "this factor definitely points toward denial" because "of the substantial flaws in the [P]etition's challenges." Prelim. Resp. 58. Patent Owner argues that "the relative size and stature of the parties also weighs in favor of denial," asserting that "[P]etitioner has nearly one-quarter of a million employees worldwide and annual revenue over $225 billion, whereas [Patent Owner] is a single-employee company." *Id.* Patent Owner asserts that Petitioner has a "strategy of filing multiple IPR petitions timed such that major deadlines in the IPRs would coincide or interfere with major deadlines in the [Texas] litigation." *Id.* at 58–59. Finally, Patent Owner argues that "there is limited public interest in the validity of the '017 Patent." *Id.* at 60.

Petitioner replies that Patent Owner's description of its challenge to claims 6–17 based on the combination of Rothschild and Anderson ("Challenge 2") as being flawed is incorrect, making several assertions as to why Patent Owner's arguments regarding the merits of Challenge 2 are unpersuasive. Prelim. Reply 1–2. Petitioner makes similar arguments regarding its challenges to claims 6–17 that use Fuller as the primary asserted reference, again making arguments responsive to Petitioner's criticisms of the merits of Patent Owner's challenges. *Id.* at 3–4. Additionally, Petitioner replies that the Texas litigation is funded by an investor. *Id.* at 5 (citing Ex. 1028).

Patent Owner replies that Petitioner's "Reply uses almost four of its five pages to respond to [Patent Owner's Preliminary Response] arguments against the merits of the challenges," which "exceeds the scope of the additional briefing requested by the [P]etitioner and granted by the Board." Prelim. Sur-reply 2–3.

As a preliminary matter, we agree that Petitioner's arguments regarding the merits of its challenges are outside the authorized scope of the Preliminary Reply. In our communication authorizing the additional briefing, we instructed the parties that "[t]he [Preliminary] Reply is limited . . . to addressing the *Fintiv* issues identified by Petitioner below." Ex. 3002, 1. Petitioner identified those issues as "Patent Owner's positions on *Fintiv* factors 1-4 and 6, and Patent Owner's 'additional considerations' related to *Fintiv*." *Id.* Petitioner's arguments regarding the merits of its challenges exceed the allowed scope for the Preliminary Reply, and we therefore do not consider such unauthorized arguments.

Petitioner presents five grounds challenging the '017 patent, with three of its challenges addressing the independent claims. Pet. 1. For reasons provided below, we find that Petitioner establishes a reasonable likelihood of prevailing on two of these challenges regarding independent claim 13. As discussed in more detail below, we find the evidence and arguments presented by Petitioner for these challenges persuasive on this preliminary record and sufficient to meet our standard for instituting *inter partes* review. For the reasons provided below, however, we are not persuaded that Petitioner establishes a reasonable likelihood of prevailing on its other challenges to the independent claims.

Patent Owner's assertions regarding Petitioner's "strategy" of filing petitions for *inter partes* review (*see* Prelim. Resp. 58–59) is speculative and, therefore, unavailing.

Accordingly, the facts underlying the merits of Petitioner's challenges to the '017 patent claims are neutral.

G. Holistic Assessment of the *Fintiv* Factors

On this record, after weighing all of the factors and taking a holistic view, we determine that the facts in this case that weigh against exercising discretion outweigh the facts that favor exercising discretion. Accordingly, we determine that the circumstances presented weigh against exercising our discretion under 35 U.S.C. § 314(a) to deny institution of *inter partes* review based on the Texas litigation.

Additionally, given that our analysis of the first five *Fintiv* factors does not favor exercising our discretion to deny institution, we need not consider whether the Petition presents a compelling, meritorious challenge. *Commscope Techs., LLC v. Dali Wireless, Inc.*, IPR2022-01242, Paper 23 at 4–5 (PTAB Feb. 27, 2023) (designated precedential).

III. PATENTABILITY ANALYSIS

A. Principles of Law

1. Inter Partes *Review*

"In an [*inter partes* review], the petitioner has the burden from the onset to show with particularity why the patent it challenges is unpatentable." *Harmonic Inc. v. Avid Tech., Inc.*, 815 F.3d 1356, 1363 (Fed. Cir. 2016) (citing 35 U.S.C. § 312(a)(3) (requiring *inter partes* review petitions to identify "with particularity . . . the evidence that supports the grounds for the challenge to each claim")). Petitioner bears the burden of persuasion to prove unpatentability of each challenged claim by a preponderance of the evidence. 35 U.S.C. § 316(e). This burden never shifts to Patent Owner. *Dynamic Drinkware, LLC v. Nat'l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015). The Board may authorize an *inter*

partes review if we determine that the information presented in the Petition and Patent Owner's Preliminary Response shows that there is a reasonable likelihood that Petitioner will prevail with respect to at least one of the claims challenged in the petition. 35 U.S.C. § 314(a).

2. Anticipation

"Under 35 U.S.C. § 102 a claim is anticipated 'if each and every limitation is found either expressly or inherently in a single prior art reference.'" *King Pharm., Inc. v. Eon Labs, Inc.*, 616 F.3d 1267, 1274 (Fed. Cir. 2010) (quoting *Celeritas Techs. Ltd. v. Rockwell Int'l Corp.*, 150 F.3d 1354, 1360 (Fed. Cir. 1998)). "Anticipation requires the presence in a single prior art disclosure of all elements of a claimed invention arranged as in the claim." *Therasense, Inc. v. Becton, Dickinson & Co.*, 593 F.3d 1325, 1332 (Fed. Cir. 2010) (quoting *Connell v. Sears, Roebuck & Co.*, 722 F.2d 1542, 1548 (Fed. Cir. 1983)).

A reference inherently discloses an element of a claim "if that missing characteristic is necessarily present, or inherent, in the single anticipating reference." *Schering Corp. v. Geneva Pharm.*, 339 F.3d 1373, 1377 (Fed. Cir. 2003) (citing *Cont'l Can Co. USA, Inc. v. Monsanto Co.*, 948 F.2d 1264, 1268 (Fed. Cir. 1991)). "Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing *may* result from a given set of circumstances is not sufficient." *Therasense*, 593 F.3d at 1332 (quoting *Cont'l Can*, 948 F.2d at 1269).

3. Obviousness

A patent claim is unpatentable under 35 U.S.C. § 103(a) if the differences between the claimed subject matter and the prior art are such that

the subject matter, as a whole, would have been obvious at the time the invention was made to a person having ordinary skill in the art to which the subject matter pertains. *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations including (1) the scope and content of the prior art, (2) any differences between the claimed subject matter and the prior art, (3) the level of skill in the art, and (4) when in evidence, any objective evidence of nonobviousness.[5] *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966).

B. Level of Ordinary Skill in the Art

The level of ordinary skill in the art is "a prism or lens" through which we view the prior art and the claimed invention. *Okajima v. Bourdeau*, 261 F.3d 1350, 1355 (Fed. Cir. 2001). The person of ordinary skill in the art is a hypothetical person presumed to have known the relevant art at the time of the invention. *In re GPAC Inc.*, 57 F.3d 1573, 1579 (Fed. Cir. 1995). In determining the level of ordinary skill in the art, we may consider certain factors, including: "(1) the educational level of the inventor; (2) type of problems encountered in the art; (3) prior art solutions to those problems; (4) rapidity with which innovations are made; (5) sophistication of the technology; and (6) educational level of active workers in the field." *Best Med. Int'l, Inc. v. Elekta Inc.*, 46 F.4th 1346, 1353 (Fed. Cir. 2022) (citations omitted). "The patent's purpose can also be informative." *Id.*

[5] At this stage of the proceeding, the parties have not directed us to any such objective evidence.

Petitioner contends that a person having ordinary skill in the art at the time of the invention "would have had a Bachelor's degree in electrical engineering, computer engineering, computer science, or a related field and two years of practical experience in technologies involved with categorizing media including images and applying metadata to such images." Pet. 7 (citing Ex. 1002 ¶ 62). Petitioner asserts that such an artisan would have had "familiarity with software for automatically generating metadata from audio and images (*e.g.*, speech-to-text and image recognition)," performed "work done in ensuring associations were maintained between metadata and images," and "standardized techniques for associating metadata with images and various known data 'container' formats." *Id.* (citing Ex. 1002 ¶ 62).

Patent Owner does not contest Petitioner's proffered definition of the level of ordinary skill in the art or offer a definition of its own. *See generally* Prelim. Resp.

We find Petitioner's description to be consistent with the problems and solutions disclosed in the '017 patent and prior art of record, and adopt it as our own for purposes of this Decision. *See, e.g.*, *In re GPAC Inc.*, 57 F.3d at 1579 (approving the determination of the level of ordinary skill in the art by appeal to the references of record).

C. Claim Construction

In an *inter partes* review, claims are construed using the same claim construction standard that would be used to construe the claims in a civil action under 35 U.S.C. § 282(b), including construing the claims in accordance with the ordinary and customary meaning of such claims as understood by one of ordinary skill in the art and the prosecution history

pertaining to the patent. 37 C.F.R. § 42.100(b). "[T]he ordinary and customary meaning of a claim term is the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention" and "after reading the entire patent." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1313, 1321 (Fed. Cir. 2005) (en banc). In addition to the specification and prosecution history, we also consider use of the terms in other claims and extrinsic evidence including expert and inventor testimony, dictionaries, and learned treatises, although extrinsic evidence is less significant than the intrinsic record. *Id.* at 1312–17. Usually, the specification is dispositive, and it is the single best guide to the meaning of a disputed term. *Id.* at 1315.

"The Board is required to construe 'only those terms . . . that are in controversy, and only to the extent necessary to resolve the controversy.'" *Realtime Data, LLC v. Iancu*, 912 F.3d 1368, 1375 (Fed. Cir. 2019) (alteration in original) (quoting *Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999)).

Petitioner contends that "[n]o formal claim constructions are needed." Pet. 2.

Patent Owner does not propose a definition for any claim term. *See generally* Prelim. Resp.

For purposes of this Decision, and based on the record before us, we determine that no construction of any term is necessary. We note that this determination does not preclude the parties from arguing their proposed constructions of the claims during trial. Indeed, the parties are hereby given notice that claim construction, in general, is an issue to be addressed at trial.

A final determination as to claim construction will be made at the close of the proceeding, after any hearing, based on all the evidence of record.

D. Overview of the Asserted Prior Art

1. Rothschild

Rothschild relates to "devices, systems and methods for embedding and retrieving information in digital images and using the information to organize, process and control the digital images." Ex. 1007 ¶ 3. Rothschild recognizes that cataloging photographs manually is an extensive process that requires a significant amount of time, and that this problem has been exacerbated by the advent of digital cameras. *Id.* ¶¶ 5–6. Rothschild purports to improve upon the prior art by providing a system and method of organizing, controlling, and manipulating digital images. *Id.* ¶ 9.

Rothschild discloses a device, such as a digital camera or mobile phone, for capturing images. Ex. 1007 ¶¶ 25, 42. Figures 1A and 1B illustrate an embodiment of the device and are reproduced below.

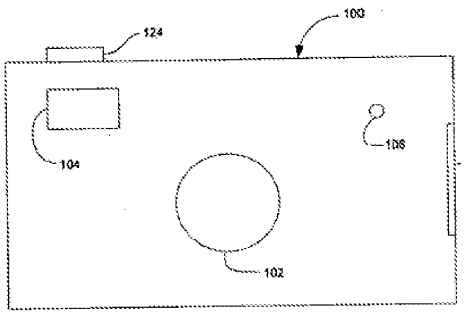


FIG. 1A

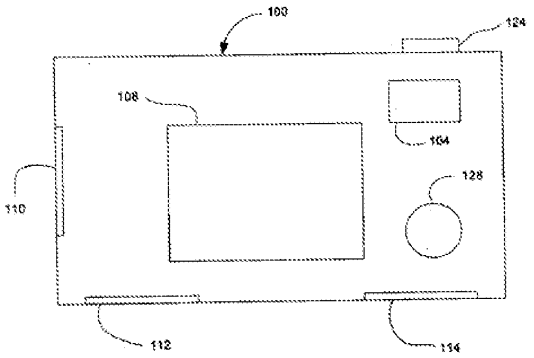


FIG. 1B

Figures 1A and 1B show front and rear views, respectively, of device 100. *Id.* ¶¶ 25–26. The device includes lens 102 for use in capturing an image, viewfinder 104, and "microphone 106 for acquiring audio, from the user of the device or from the subject of the image, which may be associated with the image." *Id.* ¶ 25. Display module 108 displays a captured image and facilitates "user input of information to be associated with a digital image." *Id.* ¶ 26. The device also includes "storage module 110 for storing a plurality of images" and "transmission module 112 for transmitting the plurality of images to another device." *Id.*

In use, the device user takes several pictures. Ex. 1007 ¶ 47. A capture module captures the images. *Id.* ¶ 29. The capture module includes an analog-to-digital (A/D) converter that converts the images into digital form. *Id.* The digital images may be processed, such as by enhancing and compressing the images, and the processed images are stored in a storage module. *Id.* The user can enter information about the images using an input module. *Id.* ¶ 47. The input module accepts user input via touch screen that overlays display module 108 or via microphone 106. *Id.* ¶ 31. The microphone may be coupled to an A/D converter that converts the user's spoken audio information into a digital format. *Id.* Furthermore, the input module may include a voice recognition processor that translates the digital human voice into alpha numeric characters for user input. *Id.* For example, for a picture of the user's "baby in Las Vegas," the user may "record an audio file indicating: 'this is a picture of my baby girl Samantha here in Las Vegas. Boy is she cute.'" *Id.* ¶¶ 47, 49. The device obtains time and date information, user identification information, and the number of the digital image, and stores this information with the audio file either as a separate

information file associated with the image or appended to the digital image file. *Id.* ¶¶ 51–52.

The images and associated files are then transferred to the user's computer, which sorts and stores the images and associated information in the location specified by the user. Ex. 1007 ¶¶ 53–54. The computer parses the associated information and stores the data in a relational database. *Id.* ¶ 63, Fig. 7. The computer may be used to print an image with symbology that encodes the location of the associated information. *Id.* ¶ 39. The device can be scanned over the printed image to read the symbology and retrieve the associated information. *Id.* ¶ 57. This allows, for example, the device to retrieve and play the audio file. *Id.* ¶ 58.

The device may include a user verification module for verifying an identity of the device user. Ex. 1007 ¶ 33. The user verification module may include an identify capture device, such as a facial recognition reader. *Id.* The identity capture device collects information from the user and compares the collected information to information that has previously been stored in the device memory. *Id.* For example, the capture module may capture an image of the user's face, which may then be processed by a facial recognition algorithm executed by the user verification module. *Id.*

2. Anderson

Anderson relates to categorizing, storing and presenting digital images. Ex. 1008, 2:42–43. Anderson's system allows "a user to categorize images at the time of capture with multiple categories of information by merely speaking into the camera." *Id.* at 2:16–19. "[T]he user's voice annotations are automatically recognized, translated, and stored in a database

. . . ." *Id.* at 2:19–22. Figure 1 illustrates an exemplary system and is reproduced below.

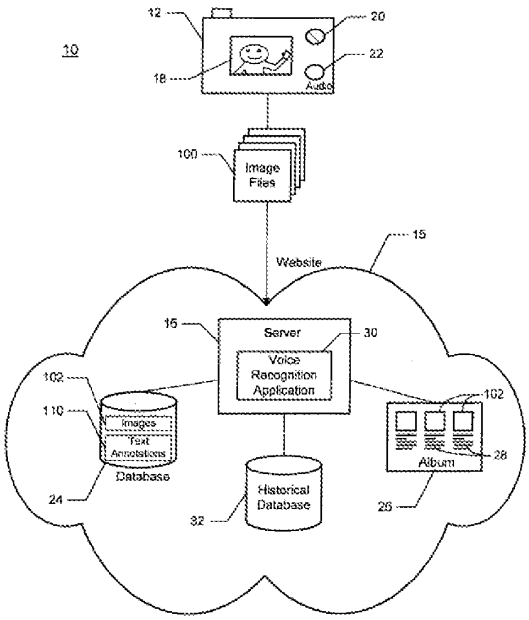


FIG. 1

Figure 1 shows system 10, including digital camera 12 for capturing digital images. *Id.* at 2:56–57. The digital camera includes microphone 20 and audio button 22 for recording sound. *Id.* at 2:58–61. A user may upload image files 100 to website 15, where web server 16 includes voice recognition application 30 and database 24. *Id.* at 2:61–65. In use, a user may record category-specific voice annotations on the digital camera, which "are then translated by voice recognition into text annotations." *Id.* at 3:10–15. An image and its corresponding text annotations are stored and indexed in database 24 "so that the images may be found more easily." *Id.* at 3:15–19.

3. Fuller

Fuller relates to the automatic extraction of metadata descriptions of digital multimedia content such as still images and video. Ex. 1003, 2:53–55. Figure 1 illustrates "a generic digital capture system . . . combined with an embedded, real-time, content-based analysis engine and data output" and is reproduced below. *Id.* at 4:49–52.

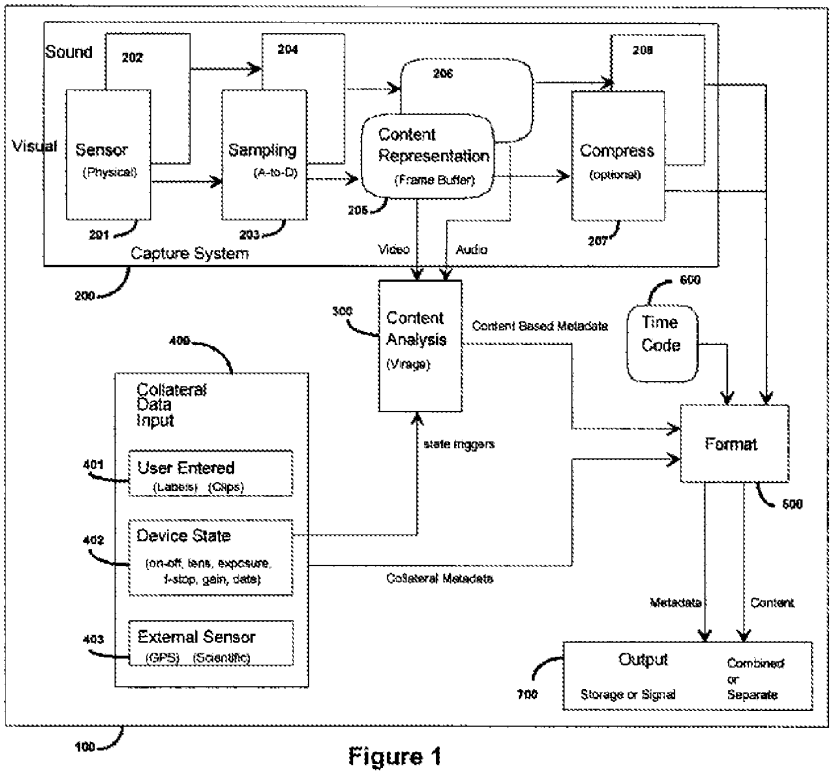


Figure 1

Figure 1 shows the overall system architecture digital capture system 100, which includes visual and audio capture subsystem 200, content-based metadata generation engine 300, collateral data gathering block 400, data formatting unit 500 with optional time code marking 600, and data output and/or storage 700. *Id.* at 5:23–29.

Visual and audio capture subsystem 200 may include visual sensor 201 and microphone 202. Ex. 1003, 5:32–34. The output from these sensors is converted to digital form and may be filtered by analog-to-digital units 203 (visual), 204 (audio), and the digital content is stored in memory units 205, 206, respectively. *Id.* at 5:34–38. The digital content may then optionally be compressed prior to being formatted at unit 500 for output or storage. *Id.* at 5:38–42.

Content-based analysis engine 300 accesses the digital representation of visual and audio content from memory units 205, 206 and extracts metadata from the content. Ex. 1003, 5:43–45. The engine performs "sophisticated analysis of multimedia content and generate[s] metadata descriptions that can be effectively used to index the content for downstream applications such as search and browse." *Id.* at 2:58–62. The content-based analysis engine may be "the Virage audio and video engines" that "are further described in U.S. [P]atent [A]pplication[] . . . No. 09/134,500" (Ex. 1005, "the '500 application"). *Id.* at 7:53–59. This engine includes a speech transcription module that processes speech and that may output recognized keywords or the full text of the transcription process, which may be used directly as metadata. Ex. 1005, 15:24–16:9.[6]

Collateral data gathering block 400 gathers various forms of collateral metadata (metadata that is not based on the content), including user-provided information at block 401, device state information at block 402, and external sensor information at block 403. Ex. 1003, 5:49–57. "Content-based metadata from the analysis engine 300 is joined by collateral metadata (not

[6] We cite to the application pagination rather than the exhibit pagination.

based on the digital content) from block 400 in the formatting unit 500." *Id.* at 5:58–60. The metadata and related digital content may be combined or stored separately. *Id.* at 6:13–34.

4. Camahan

Camahan relates to "a metabase specifically designed for the organization, management and manipulation of digital media assets." Ex. 1010 ¶ 2. Camahan's metabase is "a collection of node objects which can be implemented as XML elements and organized in a tree-like or hierarchical structure that emanates from a single root node, and stored in disk drive storage or internal cache storage." *Id.* ¶ 90. In the metabase, a "folder represents a generic container," and "[e]ach folder 201 may contain child folders 205, 207, 209 allowing the metabase to be organized in a hierarchy similar to a conventional file system." *Id.* ¶ 92; *see also id.* at Fig. 2. Further, a folder "may also contain one or more binders," where each binder "represents a media asset within the metabase and 'binds' together all of the metadata related to that asset." *Id.* ¶ 96. A binder, in turn, may contain "one or more content objects each describing a different aspect . . . of the media asset." *Id.* ¶ 100. For example, "[a] label object 211 contains structured metadata that describes the entire media asset such as the title, rating, author, etc.," and "[a] track object 213 contains structured metadata that occur at specific points or during specific intervals within the media asset such as closed captions, key frames, or speech-to-text extraction." *Id.* ¶¶ 104, 107.

5. Davis

Davis relates to associating "data with a media signal by encoding the data, a link to the data, or a combination of both into the media signal." Ex. 1013, code (57). Davis's system associates data in images or other media signals using steganographic and other techniques. *Id.* ¶ 24. The data, which may describe, for example, objects in the image or the context of the image, can be encoded within the image or stored elsewhere. *Id.* ¶¶ 15–23, 25. For example, the image can be uploaded to an image database and the metadata can be uploaded to a metadata database. *Id.* ¶ 89. Alternatively, the metadata can be stored in a database integrated with the image database. *Id.* ¶ 173.

E. Asserted Obviousness Based on Rothschild and Anderson

Petitioner argues that claims 6–17 would have been obvious in view of Rothschild and Anderson. Pet. 27–41. In support of its showing, Petitioner relies upon the Bederson Declaration. *Id.* (citing Ex. 1002). We have reviewed Petitioner's assertions and supporting evidence. For the reasons discussed below, and based on the record before us, we are not persuaded that Petitioner demonstrates a reasonable likelihood of prevailing in showing that at least one challenged claim would have been obvious in view of the combination of Rothschild and Anderson.

Independent claim 6 recites, in relevant part, "a media data converter for converting the received set of captured information to convert the received digital audio to a text based searchable file as a text context tag." Ex. 1001, 11:25–28. Independent claim 13 contains a substantially similar recitation. *Id.* at 12:34–35. As noted above, the '017 patent defines "context

tags" as searchable elements derived from the different types of data. Ex. 1001, 5:39–41. Petitioner maps Rothschild's input module 124, auxiliary input computer module 126, and user verification module 132, collectively, to the recited media data converter. Pet. 31–32. Petitioner argues that "Rothschild's input module includes [a] voice recognition processor that converts speech to text" and "Rothschild's user input data can be stored as a file associated with the captured image." *Id.* at 32–33 (citing Ex. 1007 ¶ 32; Ex. 1002 ¶ 177). Petitioner asserts that "[t]ext derived from voice recognition is a text context tag because . . . it is a searchable element derived from a context description element (*i.e.*, speech 'regarding the pictures')." *Id.* at 33 (citing Ex. 1007 ¶ 47; Ex. 1002 ¶ 177). Petitioner relies on "Jim Jones," shown in Figure 7, as an example of a context tag. *Id.* Continuing, Petitioner argues that it would have been obvious to an ordinarily skilled artisan "to use Rothschild's input module to convert received digital audio to a text based searchable file as a text context tag because doing so would allow for descriptive information to be associated with the image for later search and retrieval using a text-based search query . . . as taught by Anderson." *Id.* (citing Ex. 1002 ¶ 180).

Patent Owner refers to this challenge as "challenge 2" and the challenge to disclaimed claims 1–5 as "challenge 1," and argues that Rothschild does not disclose converting a digital audio file into a text-based searchable file. Prelim. Resp. 29–38. First, Patent Owner argues that "[t]he [P]etition does not clearly identify what is the 'external audio information' or corresponding 'stored digital audio.'" *Id.* at 30. Patent Owner asserts that "[i]t appears that the [P]etition and Dr. Bederson assume that the 'external audio information' in challenge 2 is the same as in challenge 1." *Id.* at 31

(citing Ex. 1002 ¶ 165). Patent Owner argues that this mapping of the recited external audio information fails because "there is no disclosure in Rothschild that the user of the camera speaks in association with the user-verification selfie." *Id.* at 29; *see also id.* at 31 (stating that "there is no indication in Rothschild that the user . . . speaks his name (or anything else) to the camera in connection with user verification" and "Rothschild does *not* disclose voice recognition as a user-verification technique").

Rothschild discloses a user entering information via voice recognition software regarding pictures taken by the user. Ex. 1007 ¶ 47. However, Petitioner does not map these "pictures" to the recited image. Rather, Petitioner relies on an image of the device user taken for verifying the user's identity. Pet. 28 (citing Ex. 1007 ¶ 33). For the recited audio information, Petitioner relies on its showing made for challenge 1 (*id.*), in which Petitioner maps "audio 'from the user of the device or from the subject of the image'" to the recited audio information (*id.* at 13 (citing Ex. 1007 ¶¶ 25, 31, 42, 49, Figs. 1A, 1B, 3A, 3B)). Thus, the Petition relies on audio acquired from the user or subject (*id.*) that is first converted to digital form (*id.* at 14) and then converted to text (*id.* at 32–33). But this audio is not associated with the user verification image, which the Petition maps to the recited image. *Id.* at 28–29. Neither Petitioner nor its declarant appears to explain adequately how Rothschild discloses a user entering speech regarding the user verification image. Thus, Rothschild does not appear to support Petitioner's assertions.

Independent claim 6 also recites, in relevant part, "a media data converter for . . . creating an image recognition searchable context tag with image recognition of at least a portion of the digital image." Ex. 1001,

11:25–30. Independent claim 13 contains a substantially similar recitation. *Id.* at 12:34–38. Petitioner argues that Rothschild's user verification module 132 "uses facial recognition to identify users of the device and then stores a tag associated with that user—*e.g.*, 'John Jones' (*see* FIG. 7) to associate it with images captured by that user." Pet. 35 (citing Ex. 1007 ¶¶ 33, 66, 83–84). Petitioner argues that "[t]he user's name is an image recognition context tag because it is derived from an image." *Id.*

Patent Owner argues that "nothing in Rothschild states or implies that the name of the recognized individual in the user-verification image is used as 'an image recognition searchable context tag.'" Prelim. Resp. 39. Patent Owner argues that "Rothschild does not disclose that a user image taken for user verification is treated like all the other images." *Id.*

Petitioner's mapping of the recited image here ("images captured by that user" (Pet. 35)) is different than and incompatible with Petitioner's prior mapping of the recited image ("an image of the face of the user" (*id.* at 29)). Thus, Petitioner appears to fail to explain adequately how Rothschild supports its contentions.

Additionally, neither Petitioner nor its declarant appears to explain adequately how Rothschild stores the user's name in association with the verification picture. Rothschild discloses that its "user verification module 132 will indicate and verify the identity of the user of the device 100 at the time of the image capture." Ex. 1007 ¶ 33. This verification may use a "facial recognition reader" that "will collect information on the user to be compared to information that has previously been stored in the device's memory." *Id.* "[C]apture module 122 may capture am image of the face of the user which may then be processed by a facial recognition algorithm

executed by the user verification module 132." *Id.* However, Rothschild does not appear to disclose storing a context tag associated with that user's verification image as asserted by Petitioner. *See* Pet. 35. Petitioner's reference to Rothschild's locational image verification device to transmit the user's name in association with a photograph taken by the user (*id.* (citing Ex. 1007 ¶¶ 83–84)) also fails to explain adequately how a tag is associated with the user verification image.

For at least these reasons, we question whether Petitioner demonstrates a reasonable likelihood of prevailing in showing that independent claims 6 and 13 and their dependent claims 7–12 and 14–17 would have been obvious in view of Rothschild and Anderson.

F. Assertions Based on Fuller

Petitioner argues that claims 6–9 and 13–17 are anticipated by Fuller or are rendered obvious based on the combined teachings of Fuller and Camahan. Pet. 41–57. In support of its showing, Petitioner relies upon the Bederson Declaration. *Id.* (citing Ex. 1002). We have reviewed Petitioner's assertions and supporting evidence. For the reasons discussed below, and based on the record before us, we are not persuaded that Petitioner demonstrates a reasonable likelihood of prevailing in showing that independent claim 6 is anticipated by Fuller or rendered obvious in view of Fuller and Camahan. However, we are persuaded that Petitioner demonstrates a reasonable likelihood of prevailing in showing that independent claim 13 is anticipated by Fuller or rendered obvious in view of Fuller and Camahan.

1. Independent Claim 6

Independent claim 6 recites, in relevant part, "a combiner for generating an association between the stored digital audio and the stored digital image." Ex. 1001, 11:22–24. Petitioner maps Fuller's digital audio information stored in "'memory unit' 206" to the recited stored digital audio and Fuller's "digitized image" stored in "memory unit 205" to the recited stored digital image. Pet. 45–46. Petitioner maps Fuller's formatting unit 500 to the recited combiner and argues that "the data . . . captured by Fuller's collateral data gathering block" (that is, collateral metadata) is "joined with content-based metadata in the formatting unit 500," and the joined content-based metadata and collateral metadata are stored in storage 700 with the image and audio information. *Id.* at 46–47.

However, Petitioner does not appear to explain adequately how Fuller's formatting unit 500 (that is, the Petitioner-defined combiner) creates an association between the audio information stored in memory 206 (that is, the Petitioner-defined stored digital audio) and the image information stored in memory 205 (that is, the Petitioner-defined stored digital image) as recited in claim 6. Petitioner's discussion of Fuller's collected collateral metadata and content-based metadata (Pet. 46–47) does not address this recitation of claim 6, which is directed to the stored digital audio and stored digital image rather than metadata. Petitioner's discussion of storing the combined metadata and digital content in storage 700 similarly fails to address this recitation of claim 6. Petitioner's reliance on Camahan to teach storing Fuller's keywords in XML files (Pet. 55–57) does not remedy this shortcoming.

For at least this reason, we question whether Petitioner demonstrates a reasonable likelihood of prevailing in showing that independent claim 6 and its dependent claims 7–12 are anticipated by Fuller or would have been obvious based on the combined teachings of Fuller and Camahan.

2. Independent Claim 13

a. The Preamble

Claim 13 recites "[a] system for capturing image and audio information for storage." Ex. 1001, 12:20–21. Petitioner argues that Fuller discloses such a system because it stores digital media content and metadata. Pet. 41–42 (citing Ex. 1003, 4:36–46, 5:22–25, 8:35–37, Fig. 6).

Patent Owner does not contest this aspect of the Petition. *See generally* Prelim. Resp.

Fuller discloses a digital capture system that includes a visual and audio capture subsystem. Ex. 1003 5:23–28.

Accordingly, for the foregoing reasons and on this record, to the extent the preamble is limiting, Fuller supports Petitioner's contentions.

b. The Storage Recitation

Claim 13 recites "internal storage." Ex. 1001, 12:22. Petitioner maps Fuller's memory units 205, 206, storage 700, and internal storage unit 702, collectively, to the recited internal storage. Pet. 42–43 (citing Ex. 1003 5:23–28, 5:34–38, 6:23–31, 7:35–38, 8:8–12; Ex. 1002 ¶ 200).

Patent Owner does not contest this aspect of the Petition. *See generally* Prelim. Resp.

Fuller discloses that its digital capture system 100 stores digital visual data in memory unit 205, stores digital audio data in memory unit 206, and

stores formatted digital content in storage 700. Ex. 1003 5:36–40. In an embodiment of a video camera capture device, storage 700 is referred to as internal storage unit 702. *Id.* at 8:8–10, Fig. 4.

Accordingly, for the foregoing reasons and on this record, Fuller supports Petitioner's contentions.

c. The Microphone Recitation

Claim 13 recites "a microphone interfacable with an external audio information source that generates external audio information and a first data converter for capturing the first external audio information from the microphone." Ex. 1001, 12:23–26. Petitioner maps Fuller's microphone 202 to the recited microphone and analog-to-digital unit 204 to the recited first data converter. Pet. 43 (citing Ex. 1003, 5:29–35, 7:34–37).

Patent Owner does not contest this aspect of the Petition. *See generally* Prelim. Resp.

Fuller's digital capture system 100 includes "a microphone 202 if audio is being captured." Ex. 1003, 5:32–34. Fuller discloses that the output of microphone 202 is converted to digital form by analog-to-digital unit 204. *Id.* at 5:34–36.

Accordingly, for the foregoing reasons and on this record, Fuller supports Petitioner's contentions.

d. The Camera Recitation

Claim 13 recites "a camera interfacing with an image source to capture an image therefrom." Ex. 1001, 12:27–28. Petitioner maps Fuller's digital capture system to the recited camera, noting that it may include visual

sensor 201. Pet. 44 (citing Ex. 1003, 4:49–51, 5:29–32, 7:31–34, 8:16–18; Ex. 1002 ¶¶ 203–204).

Patent Owner does not contest this aspect of the Petition. *See generally* Prelim. Resp.

Fuller's digital capture system 100 includes visual and audio capture subsystem 200, which includes visual sensor 201 that captures images. Ex. 1003, 5:30–34, 7:33–35.

Accordingly, for the foregoing reasons and on this record, Fuller supports Petitioner's contentions.

e. The First Data Converter Recitation

Claim 13 recites "the first data converter processing the captured external audio information and storing it in a first digital audio format as stored digital audio within the capture device." Ex. 1001, 12:29–31. Petitioner argues that Fuller's analog-to-digital unit 204 "processes the captured . . . audio information by digitizing it" and "stores the captured external audio information in internal storage in the capture device by outputting it to a 'sound buffer 206.'" Pet. 45 (citing Ex. 1003, 5:35–37, 7:40–43, 7:51, Figs. 1, 4–5; Ex. 1002 ¶ 205).

Patent Owner first argues that Petitioner fails to show the recited processing in Fuller. Prelim. Resp. 46. According to Patent Owner, "[w]hile Fuller does teach that 'audio is captured by a microphone 202,' that is digitized audio that is captured." *Id.* (citing Ex. 1003, 7:34–35).

Fuller's microphone 202 captures audio information in the form of sound waves, analog-to-digital unit 204 processes the captured sound into digital format, and memory 206 stores the digitized content. Ex 1003,

5:30–38, 7:35–38. The "digitization of signals from the microphone" noted by Patent Owner (*see* Prelim. Resp. 46) is the "processing." *See* Pet. 45. Petitioner's mapping of digitization to the recited processing is consistent with the processing disclosed in the '017 patent. *See, e.g.*, Ex. 1001, 6:45–48 ("One method for collecting the audio information, for example, is to utilize a microphone 210 that will capture the information and then digitize it in a digitizing block 212 utilizing an analog-to-digital converter, for example.").

Next, Patent Owner argues that Fuller does not disclose storing audio information in a first digital audio format. Prelim. Resp. 47–48. Patent Owner acknowledges that Fuller discloses digitizing audio information "in 'a sequence of 8-bit or 16-bit waveform samples at a suitable sampling frequency, such as 44.1 kHz (for CD quality audio),'" but argues that this "is not a disclosure of 'a first digital audio format.' Quite to the contrary, such a sequence of samples is just that – a sequence of samples, sans any format." *Id.* at 47 (quoting Ex. 1003, 7:40–43).

As noted above, Fuller discloses that its analog-to-digital unit 204 converts the captured audio information into digital form and stores it in memory unit 206. Ex. 1003, 5:30–38. We understand Fuller's disclosure of "digital form" to be equivalent to a digital format. Indeed, it is difficult to envision how digital audio information can be stored in memory if it is not in some digital format. That Fuller might not disclose explicitly what that format is does not indicate that there is no such format.

Accordingly, for the foregoing reasons and on this record, Fuller supports Petitioner's contentions.

f. The Processing Recitation

Claim 13 recites "the camera for processing the captured image and storing it as a stored digital image." Ex. 1001, 12:31–33. Petitioner argues that Fuller's analog-to-digital unit 203 converts the output of visual sensor 201 into digital form and "[t]he digitized image 'is then stored in a memory unit 205." Pet. 45–46 (citing Ex. 1003, 5:31–38, 7:33–40, Figs. 1, 4; Ex. 1002 ¶ 206).

Patent Owner does not contest this aspect of the Petition. *See generally* Prelim. Resp.

The output of Fuller's visual sensor 201 is converted into digital form by analog-to-digital unit 203 and the digitized content is stored in memory 205. Ex 1003, 5:30–38.

Accordingly, for the foregoing reasons and on this record, Fuller supports Petitioner's contentions.

g. The Converting Recitation

Claim 13 recites "a second data converter for converting the received digital audio to a text based searchable file as a text context tag." Ex. 1001, 12:34–35. Petitioner maps Fuller's content-based analysis engine 300 to the recited second data converter and argues that it includes "the 'Virage Audio Engine' described in the incorporated '500 application." Pet. 47 (citing Ex. 1003, 2:53–3:8, 5:43–60, 7:51–59, 8:16–18, 8:51–55, Figs. 1, 4–5; Ex. 1002 ¶ 210). Petitioner argues that the '500 application discloses that the audio engine includes audio analysis feature extractor 516, which

> classifies audio input as "speech," passes the audio signal to "a speech transcription module" that interfaces "with any available Speech Recognition Engine 624" . . . , and uses a "Keyword

> Spotting Feature Extractor 640" to "selectively identify keywords of interest and produce a text output 648 limited to the keywords specified by a Domain Dictionary 642."

Id. at 48 (citing Ex. 1005, 15:6–7, 15:14–16:9, Figs. 9–10).[7] Petitioner asserts that "[t]he resulting text defines text tags including full text or keywords" that are "stored in a text-based searchable file which contains 'tracks' of metadata." *Id.* at 49 (citing Ex. 1005, 11:1–3, 11:12; Ex. 1002 ¶ 212). Petitioner also argues that the text is searchable because Fuller discloses that "the 'content-based signal analysis' is 'added to a metadata batch under construction.'" *Id.* at 50 (citing Ex. 1003, 2:53–62, 8:45–57; Ex. 1002 ¶ 213).

Petitioner also relies on Camahan to provide additional teaching regarding text context tags. Pet. 55–57. Petitioner argues that Camahan teaches "storing text-based tracks in searchable XML files." *Id.* at 55–56. Petitioner argues that Camahan discloses a metabase of node objects, one such object being a binder that includes track objects that contain structured metadata such as speech-to-text extractions stored as searchable XML files. *Id.* at 56–57 (citing Ex. 1010 ¶¶ 90, 96, 102, 107–108, 155; Ex. 1002 ¶ 232). Petitioner argues that it would have been obvious to an ordinarily skilled artisan to store Fuller's keyword tracks as searchable XML files because "Fuller expressly suggests doing so" and "files were known data containers for metadata." *Id.* at 57 (citing Pet. 47–50;[8] Ex. 1010 ¶ 155; Ex. 1002

---

[7] On page forty-eight of the Petition, Petitioner's first citation to Exhibit 1005 refers to the exhibit pagination and Petitioner's second citation refers to the application pagination. For consistency and ease of reading, we convert Petitioner's citations to the application pagination.

[8] Although Petitioner cites to the obviousness challenge based on Rothschild and Anderson (*see* Pet. 57 (citing to "§VII.B.9")), we understand this to be

¶ 233). Petitioner also argues that the proposed modification "would have merely been the simple substitution of Fuller's tracks for a track defined as a .xml file using conventional techniques that were accessible to the [ordinarily skilled artisan]." *Id.* (citing Ex. 1002 ¶ 235).

Patent Owner first argues that "Fuller does not teach that the output of the analysis engines 300 or 301 is a file" and that "the '500 application teaches that the output of the extractor 516 is full text or keywords, not files." Prelim. Resp. 44 (citing Ex. 1003, 6:24–28, 6:66–7:2, Fig. 3; Ex. 1005, 15:6–16:17, Fig. 10).

Patent Owner's arguments are not availing. The '500 application discloses that its audio analysis feature extractor passes digitized speech data to a speech transcription module that interfaces with a speech recognition engine and outputs the full text of the transcription process, which may be used directly as metadata. Ex. 1005, 15:6–16:3. Alternatively, the output can be limited to keywords recognized in the text. *Id.* at 16:3–5. The '500 application discloses that "the metadata may be output in a variety of formats such as Virage Data Format (VDF) 562, HTML 564, XML 566, SMIL 568 and other 570." *Id.* at 13:26–28. Moreover, Fuller discloses that that the metadata extracted by the engines is used for search and browse functions. *See, e.g.*, Ex. 1003, 2:53–62.

Regarding the obviousness challenge based on Fuller and Camahan, Patent Owner acknowledges that "Camahan discloses XML files," but argues that "the tracks relied on by the [P]etition are not described as files."

---

an obvious typographical mistake, with the intended citation being to the anticipation challenge based on Fuller (§ VIII of the Petition).

Prelim. Resp. 49 (citing Ex. 1010 ¶¶ 153–157, 170). According to Patent Owner, "a track is just data, not a file." *Id.*

As explained above, Petitioner relies on Camahan to teach "storing [Fuller's] text-based tracks in searchable XML files." Pet. 55–56. Patent Owner's arguments do not address the combination of Fuller and Camahan as set forth in the Petition. Moreover, as noted above, Fuller, through the '500 application, discloses storing metadata in XML files. Ex. 1005, 13:26–28.

Accordingly, for the foregoing reasons and on this record, Fuller and the combination of Fuller and Camahan support Petitioner's contentions. We determine that Petitioner sets forth reasoning with rational underpinning as to why a person having ordinary skill in the art would have combined the teachings of Fuller and Camahan. *See* Pet. 57.

h. The Creating Recitation

Claim 13 recites the "second data converter for . . . creating an image recognition searchable context tag with image recognition of at least a portion of the digital image." Ex. 1001, 12:34, 36–38. Petitioner argues that Fuller's content-based analysis engine 300 performs metadata extraction from the digital image through face and character recognition and produces a searchable context tag for the image based on the extracted metadata. Pet. 50–51 (citing Ex. 1003, 1:59–64, 2:53–3:8, 5:35–36, 5:43–45, 5:58, 8:45–57; Ex. 1002 ¶¶ 216–217).

Patent Owner does not contest this aspect of the Petition. *See generally* Prelim. Resp.

Fuller discloses that its content-based analysis engine 300 accesses the digital representation of visual content from memory unit 205 to perform metadata extraction. Ex. 1003, 5:43–45. The generated metadata is used for search and browse functions, and may include face identification/recognition and optical character recognition. *Id.* at 2:58–3:8.

Accordingly, for the foregoing reasons and on this record, Fuller supports Petitioner's contentions.

i. The Associating Recitation

Claim 13 recites the "second data converter for . . . associating the text and image recognition context tags with the digital image." Ex. 1001, 12:34, 38–39. Petitioner argues that Fuller's content-based analysis engine 300 "joins the content-based metadata with the captured data (*i.e.*, Fuller's 'collateral metadata') in the 'formatting unit 500'" in MPEG-7 format. Pet. 51–52 (citing Ex. 1003, 5:56–6:12, 8:1–5).

Patent Owner does not contest this aspect of the Petition. *See generally* Prelim. Resp.

Fuller discloses that "[c]ontent-based metadata from the analysis engine 300 is joined by collateral metadata (not based on the digital content) from block 400 in the formatting unit 500." Ex. 1003, 5:58–60. The metadata and related digital content may be combined in a variety of formats, including MPEG-7. *Id.* at 6:13–31.

Accordingly, for the foregoing reasons and on this record, Fuller supports Petitioner's contentions.

j. The Storing Recitation

Claim 13 recites "the internal storage storing the digital image in association with the text and image recognition context tags." Ex. 1001, 12:40–41. Petitioner argues that Fuller discloses storing the combined metadata and image MPEG-7 file in internal storage unit 702. Pet. 52–53 (citing Ex. 1003, 6:13–14, 8:8–18, Figs. 1, 4; Ex. 1002 ¶ 221).

Patent Owner does not contest this aspect of the Petition. *See generally* Prelim. Resp.

Fuller discloses that "[t]he formatted output data of the formatting unit 500 can be handled by the output system 700 in a variety of ways," including combing the metadata and actual digital content in MPEG-7 format and storing the file in internal memory. Ex. 1003, 6:13–31.

Accordingly, for the foregoing reasons and on this record, Fuller supports Petitioner's contentions.

k. Conclusion Regarding Independent Claim 13

For the foregoing reasons, we determine that Petitioner has established a reasonable likelihood of prevailing on its assertion that independent claim 13 is anticipated by Fuller and would have been obvious in view of the combination of Fuller and Camahan.

IV. CONCLUSION

For the foregoing reasons, we determine that the information presented establishes a reasonable likelihood that Petitioner would prevail in showing that at least one challenged claim of the '017 patent is unpatentable. At this preliminary stage, we have not made a final determination with

respect to the patentability of the challenged claims or any underlying factual and legal issues. *See TriVascular, Inc. v. Samuels*, 812 F.3d 1056, 1068 (Fed. Cir. 2016) (noting that "there is a significant difference between a petitioner's burden to establish a 'reasonable likelihood of success' at institution, and actually proving invalidity by a preponderance of the evidence at trial").

Accordingly, *inter partes* review is instituted as to all challenged, non-disclaimed claims and all proposed grounds of unpatentability. *See SAS Inst. Inc. v. Iancu*, 138 S. Ct. 1348, 1354, 1359–60 (2018); *see also PGS Geophysical AS v. Iancu*, 891 F.3d 1354, 1360 (Fed. Cir. 2018) (interpreting the statute to require "a simple yes-or-no institution choice respecting a petition, embracing all challenges included in the petition"); Patent Trial and Appeal Board Consolidated Trial Practice Guide 64 (Nov. 2019)[9] ("The Board will not institute on fewer than all claims or all challenges in a petition.").

V. ORDER

Accordingly, it is:

ORDERED that pursuant to 35 U.S.C. § 314(a), an *inter partes* review of claims 6–13 of the '017 patent is instituted with respect to the grounds set forth in the Petition; and

FURTHER ORDERED that, pursuant to 35 U.S.C. § 314(c) and 37 C.F.R. § 42.4, notice is hereby given of the institution of a trial, which commences on the entry date of this decision.

[9] *Available at* https://www.uspto.gov/TrialPracticeGuideConsolidated.

For PETITIONER:

Andrew R. Sommer
Jay Guiliano
Harry Park
Philip Hoffmann
Jenna Kuh
GREENBERG TRAURIG, LLP
sommera@gtlaw.com
jay.guiliano@gtlaw.com
parkh@gtlaw.com
hoffmannp@gtlaw.com
kuhj@gtlaw.com

For PATENT OWNER:

Matthew Phillips
Kevin Laurence
Derek Meeker
LAURENCE & PHILLIPS IP LAW
mphillips@lpiplaw.com
klaurence@lpiplaw.com
dmeeker@lpiplaw.com

John Lord
Sarah E. Spires
SKIERMONT DERBY LLP
jlord@skiermontderby.com
sspires@skiermontderby.com

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
15/272,013	09/21/2016	MICHAEL F. MALONE	MPOR-33317

CONFIRMATION NO. 8747

25883
MUNCK WILSON MANDALA L.L.P
P.O. Drawer 800889
DALLAS, TX 75380

POWER OF ATTORNEY NOTICE

OC000000137314255

Date Mailed: 02/07/2023

NOTICE REGARDING CHANGE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 01/23/2023.

- The Power of Attorney to you in this application has been revoked by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

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**.

/zretta/

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
15/272,013	09/21/2016	MICHAEL F. MALONE	MYPT.01.01.US6

CONFIRMATION NO. 8747

40672
LAW OFFICE OF BILL NAIFEH
PO BOX 803423
DALLAS, TX 75380

POA ACCEPTANCE LETTER

OC000000137314259

Date Mailed: 02/07/2023

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 01/23/2023.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

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**.

/zretta/

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent No. 9,832,017	
Title: APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL	
Inventor: Michael F. Malone	Confirmation No. 8747
Filing Date: 2016 September 21	Examiner: Ishrat I. Sherali
Attorney Docket No. MPOR-33317	Group Art Unit: 2667

TRANSMITTAL FOR STATUTORY DISCLAIMER

A statutory disclaimer of the above-referenced patent is being filed herewith. The disclaimer is signed by the Michael F. Malone, CEO of MyPort, Inc., who is authorized to make the disclaimer. A declaration attesting to his authority accompanies the disclaimer.

The $170 fee as required by 37 C.F.R. § 1.20(d) is being paid herewith.

Respectfully submitted,

Date: January 24, 2023

/Bill R. Naifeh, Reg. No. 44,962/

Registration No. 44,962

Bill R. Naifeh
Law Office of Bill Naifeh
13155 Noel Road, Suite 900
Dallas, TX 75240
972-726-9500
214-550-2542 (fax)

STATUTORY DISCLAIMER UNDER 37 C.F.R. § 2.321(a)

MyPort, Inc. hereby disclaims claims 1-5 of U.S. Patent No. 9,832,017.

MyPort, Inc. is the owner of 100% interest in the above-referenced patent, as reflected by assignments recorded as follows: Reel 039825, frame 0671; reel 043044, frame 0298; and reel 059507, frame 0960.

Date: 1-23-23

Michael F. Malone
CEO, MyPort, Inc.

DECLARATION REGARDING AUTHORITY TO SIGN ON BEHALF OF A LEGAL ENTITY

Pursuant to 37 C.F.R. § 3.73(d)(1), I, Michael F. Malone, hereby declare that I am authorized to signed documents on behalf of MyPort, Inc.

_____ Date: 1-23-03

Michael F. Malone
CEO, MyPort, Inc.

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
15/272,013	**01/24/2023 02:16:06 PM ET**	**MPOR-33317**

Title of Invention

APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	9832017
CONFIRMATION #	8747	FILED BY	April Reasoner
PATENT CENTER #	61483483	FILING DATE	09/21/2016
CUSTOMER #	25883	FIRST NAMED INVENTOR	MICHAEL F. MALONE
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	Bill Naifeh

Documents

TOTAL DOCUMENTS: 1

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
Statutory-Disclaimer-9832017.pdf	3	Statutory disclaimers per Manual of Patent Examining Procedure(MPEP) 1490.	500 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
Statutory-Disclaimer-9832017.pdf	9040C019279E1CCACFE34FA87E7B3898820AE84C8CBF083DA FABC5E0E4A9423709200EF2F56A209BAE7E6099A4E8C41044

0762519183CA6DB5DE2F01D02620A5

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 filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC PAYMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
15/272,013	01/24/2023 02:16:06 PM ET	MPOR-33317

Title of Invention

APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	9832017
CONFIRMATION #	8747	FILED BY	April Reasoner
PATENT CENTER #	61483483	AUTHORIZED BY	Bill Naifeh
CUSTOMER #	25883	FILING DATE	09/21/2016
CORRESPONDENCE ADDRESS	-	FIRST NAMED INVENTOR	MICHAEL F. MALONE

Payment Information

PAYMENT METHOD	PAYMENT TRANSACTION ID	PAYMENT AUTHORIZED BY
CARD / 2001	E20231NE17154875	April Reasoner

FEE CODE	DESCRIPTION	ITEM PRICE($)	QUANTITY	ITEM TOTAL($)
1814	STATUTORY DISCLAIMER, INCLUDING TERMINAL DISCLAIMER	170.00	1	170.00
			TOTAL AMOUNT:	$170.00

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 filing date (see 37 CFR 1.53(b)-(d)

and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
15/272,013	**01/23/2023 07:12:04 PM ET**	**MPOR-33317**

Title of Invention

APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	9832017
CONFIRMATION #	8747	FILED BY	April Reasoner
PATENT CENTER #	61479636	FILING DATE	09/21/2016
CUSTOMER #	25883	FIRST NAMED INVENTOR	MICHAEL F. MALONE
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	Bill Naifeh

Documents

TOTAL DOCUMENTS: 1

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
POA-Statement_9832017.pdf	3	Power of Attorney	1521 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
POA-Statement_9832017.pdf	B4FB131437FE9430660E6EAE1C576E66F4B7963EB56486F324 AD105909C6DBF997C7EB3DDFF54EB06B0C81B0AE89589D27 83AC830CEA7EFE64ECFD459EF72BAC

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 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/81A (02-15)
Approved for use through 03/31/2021. OMB 0651-0035
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 - POWER OF ATTORNEY OR REVOCATION OF POWER OF ATTORNEY WITH A NEW POWER OF ATTORNEY AND CHANGE OF CORRESPONDENCE ADDRESS

Patent Number	9832017
Issue Date	11/28/2017
First Named Inventor	Michael F. Malone
Title	APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTIMEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL
Attorney Docket No.	MYPT.01.01.US6

I hereby revoke all previous powers of attorney given in the above-identified patent.

☐ A Power of Attorney is submitted herewith.

OR

☒ I hereby appoint Practitioner(s) associated with the Customer Number identified in the box at right as my/our attorney(s) or agent(s) with respect to the patent identified above, and to transact all business in the United States Patent and Trademark Office connected therewith. 40672

OR

☐ I hereby appoint Practitioner(s) named below as my/our attorney(s) or agent(s) with respect to the patent identified above, and to transact all business in the United States Patent and Trademark Office connected therewith:

Practitioner(s) Name	Registration Number

Please recognize or change the correspondence address for the above-identified patent to:

☒ The address associated with the above-identified Customer Number.

OR

☐ The address associated with the Customer Number identified in the box at right:

OR

☐ Firm or Individual Name					
Address					
City		State		Zip	
Country					
Telephone		Email			

I am the:

☐ Applicant.

OR

☒ Patent owner.

Statement under 37 CFR 3.73(c) (Form PTO/AIA/96) submitted herewith or filed on _____.

SIGNATURE of Applicant or Patent Owner

Signature		Date	12-22-17
Name	Michael F. Malone	Telephone	
Title and Company	CEO of MYPORT, INC.		

NOTE. Signatures of all the applicants or patent owners of the entire interest or their representative(s) are required. If more than one signature is required, submit multiple forms, check the box below, and identify the total number of forms submitted in the blank below.

☑ A total of 1 forms are submitted.

This collection of information is required by 37 CFR 1.31, 1.32, and 1.33. The information is required to obtain or retain a benefit by the public, which is to update (and by the USPTO to process) the file of a patent or reexamination proceeding. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/AIA/96 (08-12)
Approved for use through 11/30/2020. 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 displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(c)

Applicant/Patent Owner: MYPORT, INC.

Application No./Patent No.: 9832017 Filed/Issue Date: 11/28/2017

Titled: APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/ CONT

MYPORT, INC., a Corporation

(Name of Assignee) (Type of Assignee, e.g., corporation, partnership, university, government agency, etc.)

states that, for the patent application/patent identified above, it is (choose **one** of options 1, 2, 3 or 4 below):

1. ☑ The assignee of the entire right, title, and interest.

2. ☐ An assignee of less than the entire right, title, and interest (check applicable box):

   ☐ The extent (by percentage) of its ownership interest is _____%. Additional Statement(s) by the owners holding the balance of the interest must be submitted to account for 100% of the ownership interest.

   ☐ There are unspecified percentages of ownership. The other parties, including inventors, who together own the entire right, title and interest are:

   Additional Statement(s) by the owner(s) holding the balance of the interest must be submitted to account for the entire right, title, and interest.

3. ☐ The assignee of an undivided interest in the entirety (a complete assignment from one of the joint inventors was made). The other parties, including inventors, who together own the entire right, title, and interest are:

   Additional Statement(s) by the owner(s) holding the balance of the interest must be submitted to account for the entire right, title, and interest.

4. ☐ The recipient, via a court proceeding or the like (*e.g.*, bankruptcy, probate), of an undivided interest in the entirety (a complete transfer of ownership interest was made). The certified document(s) showing the transfer is attached.

The interest identified in option 1, 2 or 3 above (not option 4) is evidenced by either (choose **one** of options A or B below):

A. ☐ An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel _____, Frame _____, or for which a copy thereof is attached.

B. ☑ A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as follows:

1. From: Michael F. Malone To: MyPort Technologies, Inc.

   The document was recorded in the United States Patent and Trademark Office at

   Reel 039825, Frame 0671, or for which a copy thereof is attached.

2. From: MyPort Technologies, Inc. To: MyPort IP, Inc.

   The document was recorded in the United States Patent and Trademark Office at

   Reel 043044, Frame 0298, or for which a copy thereof is attached.

This collection of information is required by 37 CFR 3.73(b). 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 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.

PTO/AIA/96 (08-12)
Approved for use through 11/30/2020. 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 displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(c)

3. From: MyPort IP, Inc. To: MyPort, Inc.

The document was recorded in the United States Patent and Trademark Office at
Reel 059507, Frame 0960, or for which a copy thereof is attached.

4. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at
Reel _____, Frame _____, or for which a copy thereof is attached.

5. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at
Reel _____, Frame _____, or for which a copy thereof is attached.

6. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at
Reel _____, Frame _____, or for which a copy thereof is attached.

☐ Additional documents in the chain of title are listed on a supplemental sheet(s).

☑ As required by 37 CFR 3.73(c)(1)(i), the documentary evidence of the chain of title from the original owner to the assignee was, or concurrently is being, submitted for recordation pursuant to 37 CFR 3.11.

[NOTE: A separate copy (i.e., a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, to record the assignment in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

Signature	Date: 1-23-23
Michael F. Malone	CEO
Printed or Typed Name	Title or Registration Number

AO 120 (Rev. 08/10)

TO:	REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK
Mail Stop 8 **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court Eastern District of Texas on the following ☐ Trademarks or ☒ Patents. (☐ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED	U.S. DISTRICT COURT
2:22-cv-114	4/15/2022	Eastern District of Texas

PLAINTIFF	DEFENDANT
MYPORT, INC.	SAMSUNG ELECTRONICS CO., LTD and SAMSUNG ELECTRONICS AMERICA, INC.

	PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1	9,832,017	11/28/2017	MyPort, Inc.
2	10,237,067	3/19/2019	MyPort, Inc.
3	10,721,066	7/21/2020	MyPort, Inc.
4			
5			

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY ☐ Amendment ☐ Answer ☐ Cross Bill ☐ Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director **Copy 3—Upon termination of action, mail this copy to Director**
Copy 2—Upon filing document adding patent(s), mail this copy to Director **Copy 4—Case file copy**

American LegalNet, Inc.
www.FormsWorkFlow.com

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
15/272,013	11/28/2017	9832017	MPOR-33317	8747

25883 7590 11/08/2017

HOWISON & ARNOTT, L.L.P
P.O. BOX 741715
DALLAS, TX 75374-1715

ISSUE NOTIFICATION

The projected patent number and issue date are specified above.

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment is 0 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Application Assistance Unit (AAU) of the Office of Data Management (ODM) at (571)-272-4200.

APPLICANT(s) (Please see PAIR WEB site http://pair.uspto.gov for additional applicants):

MICHAEL F. MALONE, McKinney, TX;
MYPORT TECHNOLOGIES, INC., McKinney, TX;

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 USA offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to encourage and facilitate business investment. To learn more about why the USA is the best country in the world to develop technology, manufacture products, and grow your business, visit SelectUSA.gov.

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

Application Number	15272013
Filing Date	2016-09-21
First Named Inventor	MALONE, Michael F.
Art Unit	2161
Examiner Name	
Attorney Docket Number	MPOR-33317

	174	6747692		2004-06-08	PATEL ET AL.	
	175	6670984		2003-12-30	TANAKA ET AL.	
Change(s) applied to document, /D.D./ 8/11/2017	176	6628325		09-2003 ~~2003-12-30~~	STEINBERG ET AL.	
	177	6535298		2003-03-18	WINTER ET AL	
	178	6411328		2002-06-25	FRANKE ET AL.	
	179	7095871		2006-08-22	JONES ET AL.	
	180	5841978		1998-11-24	RHOADS	
	181	5410598		1995-04-25	SHEAR	
	182	5493677		1996-02-20	BALOGH ET AL.	
	183	5502576		1996-03-26	RAMSAY ET AL.	
	184	5519778		1996-05-21	LEIGHTON ET AL.	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

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

Application Number	15272013
Filing Date	2016-09-21
First Named Inventor	MALONE, Michael F.
Art Unit	2161
Examiner Name	
Attorney Docket Number	MPOR-33317

	64	5767496		1998-06-16	SWARTZ ET AL.	
Change(s) applied to document, /D.D./ 8/11/2017	65	5,787,186 ~~5787183~~		1998-07-28	SCHROEDER	
	66	5799092		1998-08-25	KRISTOL ET AL.	
	67	5841886		1998-11-24	RHOADS	
	68	5857038		1999-01-05	OWADA ET AL.	
	69	5907149		1999-05-25	MARCKINI	
	70	5995630		1999-11-30	BORZA	
	71	6005936		1999-12-21	SCHIMIZU ET AL.	
	72	6064764		2000-05-16	BHASKARAN ET AL.	
	73	6148091		2000-11-24	DIMARIA	
	74	6185316		2001-02-06	BUFFAM	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

25883 7590 07/28/2017
HOWISON & ARNOTT, L.L.P
P.O. BOX 741715
DALLAS, TX 75374-1715

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission
I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
15/272,013	09/21/2016	MICHAEL F. MALONE	MPOR-33317	8747

TITLE OF INVENTION: APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE MATATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	UNDISCOUNTED	$960	$0	$0	$960	10/30/2017

EXAMINER	ART UNIT	CLASS-SUBCLASS
SHERALI, ISHRAT I	2667	345-732000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
- ☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
- ☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
(1) The names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 Munck Wilson Mandala, LLP
2
3

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE: MYPORT IP, INC.

(B) RESIDENCE: (CITY and STATE OR COUNTRY): McKinney, Texas

Please check the appropriate assignee category or categories (will not be printed on the patent) : ☐ Individual ☒ Corporation or other private group entity ☐ Government

4a. The following fee(s) are submitted:
- ☒ Issue Fee
- ☒ Publication Fee (No small entity discount permitted)
- ☐ Advance Order - # of Copies _____

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**
- ☐ A check is enclosed.
- ☐ Payment by credit card. Form PTO-2038 is attached.
- ☒ The director is hereby authorized to charge the required fee(s), any deficiency, or credits any overpayment, to Deposit Account Number 200780 (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
- ☐ Applicant certifying micro entity status. See 37 CFR 1.29 — NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.
- ☐ Applicant asserting small entity status. See 37 CFR 1.27 — NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.
- ☐ Applicant changing to regular undiscounted fee status. — NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature /Keith D. Harden, Reg. # 74472/ Date October 26, 2017

Typed or printed name Keith D. Harden Registration No. 74,472

PTOL-85 Part B (10-13) Approved for use through 10/31/2013. OMB 0651-0033 U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Electronic Patent Application Fee Transmittal

Application Number:	15272013
Filing Date:	21-Sep-2016
Title of Invention:	APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/ VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG (S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL
First Named Inventor/Applicant Name:	MICHAEL F. MALONE
Filer:	Keith D. Harden/Paula Sandu
Attorney Docket Number:	MPOR-33317

Filed as Large Entity

Filing Fees for Utility under 35 USC 111(a)

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
UTILITY APPL ISSUE FEE	1501	1	960	960

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Extension-of-Time:				
Miscellaneous:				
			Total in USD ($)	**960**

Electronic Acknowledgement Receipt	
EFS ID:	30772951
Application Number:	15272013
International Application Number:	
Confirmation Number:	8747
Title of Invention:	APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/ VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG (S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL
First Named Inventor/Applicant Name:	MICHAEL F. MALONE
Customer Number:	25883
Filer:	Keith D. Harden
Filer Authorized By:	
Attorney Docket Number:	MPOR-33317
Receipt Date:	27-OCT-2017
Filing Date:	21-SEP-2016
Time Stamp:	11:57:33
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	DA
Payment was successfully received in RAM	$960
RAM confirmation Number	102717INTEFSW00013804200780
Deposit Account	200780
Authorized User	Keith Harden
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows: 37 CFR 1.17 (Patent application and reexamination processing fees) 37 CFR 1.21 (Miscellaneous fees and charges)	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Issue Fee Payment (PTO-85B)	MPO33317IssueFeeTransmittal.pdf	95670 c3810cc4500680a6bc8e6047cdc5d8afa6942561	no	1
Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	30817 3c0a7619bb4be0aa5ade5a715a94ffa73a06a277	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	126487		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.
National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.
New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
15/272,013	09/21/2016	2667	1600	MPOR-33317	20	2

CONFIRMATION NO. 8747

CORRECTED FILING RECEIPT

25883
HOWISON & ARNOTT, L.L.P
P.O. BOX 741715
DALLAS, TX 75374-1715

OC000000094898407

Date Mailed: 10/19/2017

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Inventor(s)

MICHAEL F. MALONE, McKinney, TX;

Applicant(s)

MYPORT TECHNOLOGIES, INC., McKinney, TX;

Power of Attorney: The patent practitioners associated with Customer Number 25883

Domestic Priority data as claimed by applicant

This application is a CON of 14/660,166 03/17/2015 ABN
which is a CON of 13/965,625 08/13/2013 PAT 8983119
which is a CON of 13/417,229 03/10/2012 PAT 8509477
which is a CON of 12/857,358 08/16/2010 PAT 8135169
which is a CON of 11/621,062 01/08/2007 PAT 7778438
which claims benefit of 60/757,075 01/06/2006
and is a CIP of 11/325,373 01/04/2006 PAT 7184573
which is a CON of 10/674,910 09/29/2003 PAT 6996251
which claims benefit of 60/414,449 09/30/2002

Foreign Applications for which priority is claimed (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see http://www.uspto.gov for more information.) - None.
Foreign application information must be provided in an Application Data Sheet in order to constitute a claim to foreign priority. See 37 CFR 1.55 and 1.76.

Permission to Access Application via Priority Document Exchange: Yes

Permission to Access Search Results: Yes

Applicant may provide or rescind an authorization for access using Form PTO/SB/39 or Form PTO/SB/69 as appropriate.

If Required, Foreign Filing License Granted: 10/03/2016
The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 15/272,013**
Projected Publication Date: Not Applicable
Non-Publication Request: No
Early Publication Request: No
Title

APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL

Preliminary Class

345

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.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
15/272,013	09/21/2016	MICHAEL F. MALONE	MPOR-33317	8747

25883 7590 10/19/2017

HOWISON & ARNOTT, L.L.P
P.O. BOX 741715
DALLAS, TX 75374-1715

EXAMINER	
SHERALI, ISHRAT I	

ART UNIT	PAPER NUMBER
2667	

NOTIFICATION DATE	DELIVERY MODE
10/19/2017	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

patents@dalpat.com

Response to Rule 312 Communication	Application No.	Applicant(s)
	15/272,013	MALONE, MICHAEL F.
	Examiner	Art Unit
	ISHRAT I. SHERALI	2667

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address –

1. ☒ The amendment filed on *03 October 2017* under 37 CFR 1.312 has been considered, and has been:

a) ☒ entered.

b) ☐ entered as directed to matters of form not affecting the scope of the invention.

c) ☐ disapproved because the amendment was filed after the payment of the issue fee.
Any amendment filed after the date the issue fee is paid must be accompanied by a petition under 37 CFR 1.313(c)(1) and the required fee to withdraw the application from issue.

d) ☐ disapproved. See explanation below.

e) ☐ entered in part. See explanation below.

/ISHRAT I SHERALI/
Primary Examiner, Art Unit 2667

Please enter this amendment
10/12/20 EX. /IS/

MPOR-33317 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	Michel F. MALONE
Serial No.:	15/272,013
Filed:	September 21, 2016
Confirmation No.	8747
Notice of Allowance Mailed:	July 28, 2017
Group:	2667
Examiner:	Ishrat I. SHERALI
For:	METHOD FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH CONVERSION, METATAGS CREATION, STORAGE AND SEARCH RETRIEVAL

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

Dear Sir:

AMENDMENT UNDER RULE 312

The above-identified application has been carefully reviewed prior to the payment of the Issue Fee. It is requested that the following amendments be entered under the provisions of Rule 312.

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

Listin of the Claims begin on page 3 of this paper.

Remarks begin on page 8 of this paper.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	Michel F. MALONE
Serial No.:	15/272,013
Filed:	September 21, 2016
Confirmation No.	8747
Notice of Allowance Mailed:	July 28, 2017
Group:	2667
Examiner:	Ishrat I. SHERALI
For:	METHOD FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH CONVERSION, METATAGS CREATION, STORAGE AND SEARCH RETRIEVAL

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

Dear Sir:

AMENDMENT UNDER RULE 312

The above-identified application has been carefully reviewed prior to the payment of the Issue Fee. It is requested that the following amendments be entered under the provisions of Rule 312.

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

Listin of the Claims begin on page 3 of this paper.

Remarks begin on page 8 of this paper.

In the Specification

Please replace the title with the following amended title:

APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE ~~MATATAG(S)~~ METATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL

In the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

1-20. (Canceled)

21. (Previously Presented) A system for capturing image and audio information for storage in a database at a location on a network, comprising:

a microphone interfacable with an external audio information source that generates external audio information and a first data converter for capturing the first external audio information from the microphone,

a camera interfacing with an external image source to capture an image therefrom;

the first data converter processing the captured external audio information and storing it in a first digital audio format as stored digital audio within the capture device, the camera for processing the captured image and storing it as a stored digital image;

a transmitter for transmitting the stored digital audio and stored digital image to the location on the network; and

a system disposed at the location on the network and including:

a receiver for receiving the transmitted digital audio and digital image from the transmitter as a received set of captured information,

a system data converter for converting the received digital audio to a text based searchable file as a text context tag and creating an image recognition searchable context tag with image recognition of at least a portion of the digital image and associating the text and image recognition context tags with the digital image, and

the database storing the digital image in association with the text and image recognition context tags.

22. (Previously Presented) The system of Claim 21, wherein the first data converter captures the first external audio information from the microphone during generation thereof.

23. (Previously Presented) The system of Claim 21, wherein the camera captures the image from the external image source at an instant in time.

24. (Previously Presented) The system of Claim 21, wherein the first data converter processes the captured external audio information from a start event representing the time that capture of the external audio information is initiated to a stop event representing the time that capture of the external audio information is complete.

25. (Previously Presented) The system of Claim 24, wherein the transmitter transmits the stored digital audio information and the stored digital image to the location on the network after at least the stop event associated with the processing of the captured external audio information.

26. (Previously Presented) A system for capturing image and audio information for storage, comprising:

a capture device having:

internal storage;

a microphone interfacable with and external audio information source that generates external audio information and a first data converter for capturing the first external audio information from the microphone,

a camera interfacing with and external image source to capture an image therefrom;

the first data converter processing the captured external audio information and storing it in a first digital audio format as stored digital audio in internal storage within the capture device, the camera for processing the captured image and storing it as a stored digital image in internal storage and a combiner for generating an association between the stored digital audio and the stored digital image,

a media data converter for converting the received set of captured information to convert the received digital audio to a text based searchable file as a text context tag and creating an image recognition searchable context tag with

image recognition of at least a portion of the digital image and associating the text and image recognition context tags with the digital image, and

the internal storage storing the digital image in association with the text and image recognition context tags.

27. (Previously Presented) The system of Claim 26, wherein the first data converter captures the first external audio information from the microphone during generation thereof.

28. (Previously Presented) The system of Claim 26, wherein the camera captures the image from the external image source at an instant in time.

29. (Previously Presented) The system of Claim 26, wherein the first data converter processes the captured external audio information from a start event representing the time that capture of the external audio information is initiated to a stop event representing the time that capture of the external audio information is complete.

30. (Previously Presented) The system of Claim 26, and further including:

a transmitter associated with the capture device for transmitting the associated stored digital image in association with the text and image recognition context tags to the location on the network;

a system disposed at the location on a network and including:

a receiver for receiving the transmitted associated stored digital image in association with the text and image recognition context tags from the transmitter associated with the capture device as a received set of captured information,

a database, and

the database storing the received associated stored digital image in association with the text and image recognition context tags.

31. (Previously Presented) The system of Claim 30, wherein the first data converter processes the captured external audio information from a start event representing the time that

capture of the external audio information is initiated to a stop event representing the time that capture of the external audio information is complete.

32. (Previously Presented) The system of Claim 31, wherein the transmitter transmits the associated stored digital image in association with the text and image recognition context tags to the location on the network after at least the stop event associated with the processing of the captured external audio information.

33. (Previously Presented) A system for capturing image and audio information for storage, comprising:

internal storage;

a microphone interfacable with an external audio information source that generates external audio information and a first data converter for capturing the first external audio information from the microphone;

a camera interfacing with an image source to capture an image therefrom;

the first data converter processing the captured external audio information and storing it in a first digital audio format as stored digital audio within the capture device, the camera for processing the captured image and storing it as a stored digital image;

a second data converter for converting the received digital audio to a text based searchable file as a text context tag and creating an image recognition searchable context tag with image recognition of at least a portion of the digital image and associating the text and image recognition context tags with the digital image; and

the internal storage storing the digital image in association with the text and image recognition context tags.

34. (Previously Presented) The system of Claim 33, wherein the image source is an external image source.

35. (Previously Presented) The system of Claim 33, wherein the first data converter captures the first external audio information from the microphone during generation thereof.

36. (Previously Presented) The system of Claim 33, wherein the camera captures the image from the image source at an instant in time.

37. (Previously Presented) The system of Claim 33, wherein the first data converter processes the captured external audio information from a start event representing the time that capture of the external audio information is initiated to a stop event representing the time that capture of the external audio information is complete.

REMARKS

Subsequent to the receipt of a Notice of Allowance, this application was carefully reviewed prior to payment of the Issue Fee. The errors noted during this review have been corrected by this Amendment Under Rule 312. It is respectfully submitted that the foregoing amendments do not affect the merits of the application and are proper subject matter for an Amendment Under Rule 312.

Entry of this Amendment is respectfully requested.

Respectfully submitted,
MUNCK WILSON MANDALA, LLP
Attorneys for Applicant(s)

/Keith D. Harden, Reg. #74472/

Keith D. Harden
Registration No. 74,472

KDH/alb
600 Banner Place
12770 Coit Road
Dallas, Texas 75251
Tel: 972-628-3600
Fax: 972-628-3616
October 2, 2017

Electronic Acknowledgement Receipt	
EFS ID:	30544553
Application Number:	15272013
International Application Number:	
Confirmation Number:	8747
Title of Invention:	APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/ VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE MATATAG (S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL
First Named Inventor/Applicant Name:	MICHAEL F. MALONE
Customer Number:	25883
Filer:	Keith D. Harden
Filer Authorized By:	
Attorney Docket Number:	MPOR-33317
Receipt Date:	03-OCT-2017
Filing Date:	21-SEP-2016
Time Stamp:	11:28:12
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		MPO33317AMD312.pdf	102811 973567f519ab1e0b3ef103907514d12e4554c6bd	yes	8

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Amendment after Notice of Allowance (Rule 312)	1	1
	Specification	2	2
	Claims	3	7
	Applicant Arguments/Remarks Made in an Amendment	8	8
Warnings:			
Information:			
	Total Files Size (in bytes):	102811	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.
National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.
New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

25883 7590 07/28/2017
HOWISON & ARNOTT, L.L.P
P.O. BOX 741715
DALLAS, TX 75374-1715

EXAMINER	
SHERALI, ISHRAT I	
ART UNIT	PAPER NUMBER
2667	

DATE MAILED: 07/28/2017

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
15/272,013	09/21/2016	MICHAEL F. MALONE	MPOR-33317	8747

TITLE OF INVENTION: APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE MATATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	UNDISCOUNTED	$960	$0	$0	$960	10/30/2017

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the ENTITY STATUS shown above. If the ENTITY STATUS is shown as SMALL or MICRO, verify whether entitlement to that entity status still applies.

If the ENTITY STATUS is the same as shown above, pay the TOTAL FEE(S) DUE shown above.

If the ENTITY STATUS is changed from that shown above, on PART B - FEE(S) TRANSMITTAL, complete section number 5 titled "Change in Entity Status (from status indicated above)".

For purposes of this notice, small entity fees are 1/2 the amount of undiscounted fees, and micro entity fees are 1/2 the amount of small entity fees.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PTOL-85 (Rev. 02/11)

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

25883 7590 07/28/2017
HOWISON & ARNOTT, L.L.P
P.O. BOX 741715
DALLAS, TX 75374-1715

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission
I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
15/272,013	09/21/2016	MICHAEL F. MALONE	MPOR-33317	8747

TITLE OF INVENTION: APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE MATATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL.

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	UNDISCOUNTED	$960	$0	$0	$960	10/30/2017

EXAMINER	ART UNIT	CLASS-SUBCLASS
SHERALI, ISHRAT I	2667	345-732000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
- ❑ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
- ❑ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
(1) The names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.
1 _____
2 _____
3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE (B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ❑ Individual ❑ Corporation or other private group entity ❑ Government

4a. The following fee(s) are submitted:
- ❑ Issue Fee
- ❑ Publication Fee (No small entity discount permitted)
- ❑ Advance Order - # of Copies _____

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**
- ❑ A check is enclosed.
- ❑ Payment by credit card. Form PTO-2038 is attached.
- ❑ The director is hereby authorized to charge the required fee(s), any deficiency, or credits any overpayment, to Deposit Account Number _____ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
- ❑ Applicant certifying micro entity status. See 37 CFR 1.29 — NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.
- ❑ Applicant asserting small entity status. See 37 CFR 1.27 — NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.
- ❑ Applicant changing to regular undiscounted fee status. — NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature _____ Date _____

Typed or printed name _____ Registration No. _____

PTOL-85 Part B (10-13) Approved for use through 10/31/2013. OMB 0651-0033 U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
15/272,013	09/21/2016	MICHAEL F. MALONE	MPOR-33317	8747

25883 7590 07/28/2017

HOWISON & ARNOTT, L.L.P
P.O. BOX 741715
DALLAS, TX 75374-1715

EXAMINER	
SHERALI, ISHRAT I	

ART UNIT	PAPER NUMBER
2667	

DATE MAILED: 07/28/2017

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(Applications filed on or after May 29, 2000)

The Office has discontinued providing a Patent Term Adjustment (PTA) calculation with the Notice of Allowance.

Section 1(h)(2) of the AIA Technical Corrections Act amended 35 U.S.C. 154(b)(3)(B)(i) to eliminate the requirement that the Office provide a patent term adjustment determination with the notice of allowance. See Revisions to Patent Term Adjustment, 78 Fed. Reg. 19416, 19417 (Apr. 1, 2013). Therefore, the Office is no longer providing an initial patent term adjustment determination with the notice of allowance. The Office will continue to provide a patent term adjustment determination with the Issue Notification Letter that is mailed to applicant approximately three weeks prior to the issue date of the patent, and will include the patent term adjustment on the patent. Any request for reconsideration of the patent term adjustment determination (or reinstatement of patent term adjustment) should follow the process outlined in 37 CFR 1.705.

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

OMB Clearance and PRA Burden Statement for PTOL-85 Part B

The Paperwork Reduction Act (PRA) of 1995 requires Federal agencies to obtain Office of Management and Budget approval before requesting most types of information from the public. When OMB approves an agency request to collect information from the public, OMB (i) provides a valid OMB Control Number and expiration date for the agency to display on the instrument that will be used to collect the information and (ii) requires the agency to inform the public about the OMB Control Number's legal significance in accordance with 5 CFR 1320.5(b).

The information collected by PTOL-85 Part B is required by 37 CFR 1.311. 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, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450. 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.

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.

Notice of Allowability	Application No. 15/272,013	Applicant(s) MALONE, MICHAEL F.	
	Examiner ISHRAT I. SHERALI	Art Unit 2667	AIA (First Inventor to File) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *6/26/2017and 7/19/2017*.
   ☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on_____.
2. ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.
3. ☒ The allowed claim(s) is/are *21-37 (renumbered as 1-17)*. As a result of the allowed claim(s), you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.
4. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

   Certified copies:

   a) ☐ All b) ☐ Some *c) ☐ None of the:
   1. ☐ Certified copies of the priority documents have been received.
   2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
   3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☒ Notice of References Cited (PTO-892)
2. ☐ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date _____
3. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
4. ☒ Interview Summary (PTO-413), Paper No./Mail Date *7/19/2017* .
5. ☐ Examiner's Amendment/Comment
6. ☐ Examiner's Statement of Reasons for Allowance
7. ☐ Other _____.

/ISHRAT I SHERALI/
Primary Examiner, Art Unit 2667

Examiner-Initiated Interview Summary	Application No.	Applicant(s)	
	15/272,013	MALONE, MICHAEL F.	
	Examiner	Art Unit	
	ISHRAT I. SHERALI	2667	

All participants (applicant, applicant's representative, PTO personnel):

(1) *ISHRAT I. SHERALI*. (3)_____.

(2) *Keith Harden*. (4)_____.

Date of Interview: *19 July 2017*.

Type: ☒ Telephonic ☐ Video Conference
☐ Personal [copy given to: ☐ applicant ☐ applicant's representative]

Exhibit shown or demonstration conducted: ☐ Yes ☒ No.
If Yes, brief description: _____.

Issues Discussed ☐101 ☐112 ☐102 ☐103 ☒Others
(For each of the checked box(es) above, please describe below the issue and detailed description of the discussion)

Claim(s) discussed: *21-37*.

Identification of prior art discussed: *MALONE (US 7,778,438)*.

Substance of Interview
(For each issue discussed, provide a detailed description and indicate if agreement was reached. Some topics may include: identification or clarification of a reference or a portion thereof, claim interpretation, proposed amendments, arguments of any applied references etc...)

Examiner Applicant's Attorney applicant has canceld claims 1-20 and added new claims 21-37. New claims 21-37 are narrower than claims 1-12 of parent application now US Patent No. 7778438. However claims 21-37 represent the same invention as claims 1-12 of US Patent No 7778438. Since claims 1-12 of US Patent 7778438 are open ended claims because claims recite comprising therefore claims 1-12 of US Patent include limitations in instant application. Applicant's Attorney agreed to file terminal disclaimer.

Applicant recordation instructions: It is not necessary for applicant to provide a separate record of the substance of interview.

Examiner recordation instructions: Examiners must summarize the substance of any interview of record. A complete and proper recordation of the substance of an interview should include the items listed in MPEP 713.04 for complete and proper recordation including the identification of the general thrust of each argument or issue discussed, a general indication of any other pertinent matters discussed regarding patentability and the general results or outcome of the interview, to include an indication as to whether or not agreement was reached on the issues raised.

☐ Attachment

/ISHRAT I SHERALI/ Primary Examiner, Art Unit 2667	

Notice of References Cited	Application/Control No. 15/272,013	Applicant(s)/Patent Under Reexamination MALONE, MICHAEL F.	
	Examiner ISHRAT I. SHERALI	Art Unit 2667	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	CPC Classification	US Classification
*	A	US-8,768,693 B2	07-2014	Somekh; Oren	G06F17/30265	348/231.4
*	B	US-7,269,596 B2	09-2007	Williams; Michael John	H04N21/235	1/1
*	C	US-7,043,048 B1	05-2006	Ellingson; Eric E.	H04N1/32101	382/100
*	D	US-6,687,383 B1	02-2004	Kanevsky; Dimitri	G10L19/018	380/210
	E	US-				
	F	US-				
	G	US-				
	H	US-				
	I	US-				
	J	US-				
	K	US-				
	L	US-				
	M	US-				

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	CPC Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	15272013	MALONE, MICHAEL F.
	Examiner	Art Unit
	ISHRAT I SHERALI	2667

CPC- SEARCHED		
Symbol	Date	Examiner

CPC COMBINATION SETS - SEARCHED		
Symbol	Date	Examiner

US CLASSIFICATION SEARCHED			
Class	Subclass	Date	Examiner
382	100, 305	7/20/17	IS

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

SEARCH NOTES		
Search Notes	Date	Examiner
SEARCH EAST TEXT LIMITED CPC/USPC, INVENTOR/ASSINGEE, GOOGLE	7/20/17	IS
SEARCH EAST TEXT LIMITED FOLLOWING CPC CLASS/SUBCLASS: G06F 17/30038, 30749, 30268, 3028, 30244, H04L 9/14, H04N 5/76, 44, H04N 21/25816, 25875, 26603, 26613	7/20/17	IS

INTERFERENCE SEARCH			
US Class/ CPC Symbol	US Subclass / CPC Group	Date	Examiner
	SEARCH EAST INTERFERENCE TEXT LIMITED CPC/USPC	7/20/17	IS

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	15272013	MALONE, MICHAEL F.
	Examiner	Art Unit
	ISHRAT I SHERALI	2667

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☒ Claims renumbered in the same order as presented by applicant ☐ CPA ☒ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	03/20/2017				07/21/2017				
	1	-								
	2	-								
	3	-								
	4	-								
	5	-								
	6	-								
	7	-								
	8	-								
	9	-								
	10	-								
	11	-								
	12	-								
	13	-								
	14	-								
	15	-								
	16	-								
	17	-								
	18	-								
	19	-								
	20	-								
1	21	=								
2	22	=								
3	23	=								
4	24	=								
5	25	=								
6	26	=								
7	27	=								
8	28	=								
9	29	=								
10	30	=								
11	31	=								
12	32	=								
13	33	=								
14	34	=								
15	35	=								
16	36	=								

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	15272013	MALONE, MICHAEL F.
	Examiner	**Art Unit**
	ISHRAT I SHERALI	2667

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☒ Claims renumbered in the same order as presented by applicant ☐ CPA ☒ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	03/20/2017				07/21/2017				
17	37	=								

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	2928	(captur$5 or record$3) with ((audio or speech or voice) with image) and (convert$3 or conversion) with (audio or voice or speech)and(datbase or data adj2 base or librar$4 archiv$6)	USPAT	OR	ON	2017/07/20 12:57
L2	722	(captur$5 or record$3) with ((audio or speech or voice) with image) and (convert$3 or conversion) with (audio or voice or speech) with text and(datbase or data adj2 base or librar$4 archiv$6)	USPAT	OR	ON	2017/07/20 12:57
L3	524	(captur$5 or record$3) with ((audio or speech or voice) with image) and (convert$3 or conversion) with (audio or voice or speech) with text and(datbase or data adj2 base or librar$4) and search$3	USPAT	OR	ON	2017/07/20 12:59
L4	165	(captur$5 or record$3) with ((audio or speech or voice) with image) and (convert$3 or conversion) with (audio or voice or speech) with text and(datbase or data adj2 base or librar$4) same (tag$4 or metatag$6 or meta$6) and search$3	USPAT	OR	ON	2017/07/20 13:01
L5	3	(captur$5 or record$3) with ((audio or speech or voice) with image) same camera and (convert$3 or conversion) with (audio or voice or speech) with text and(datbase or data adj2 base or librar$4) same (tag$4 or metatag$6 or meta$6) same search$3	USPAT	OR	ON	2017/07/20 13:03
L6	31	(captur$5 or record$3) with ((audio or speech) with image) same camera and (convert$3 or conversion) with (audio or voice or speech) with text and(datbase or data adj2 base or librar$4) same (tag$4 or metatag$6 or meta$6) and search$3	USPAT	OR	ON	2017/07/20 13:05
L7	89	(captur$5 or record$3) with ((audio or speech) with image) same camera and (convert$3 or conversion) with (audio or voice or speech) with text and(datbase or data adj2 base or librar$4) same (tag$4 or metatag$6 or meta$6) and search$3	US-PGPUB; USPAT	OR	ON	2017/07/20 13:07
L8	89	(captur$5 or record$3) with ((audio or speech) with image) same camera and (convert$3 or conversion) with (audio or voice or speech) with text and(datbase or data adj2 base or librar$4) same (tag$4 or metatag$6 or meta$6) and (quer$3 or search$3 or retriv$6)	US-PGPUB; USPAT	OR	ON	2017/07/20 13:08
L9	0	(captur$5 or record$3) with ((audio or speech) with image) same camera and (convert$3 or conversion) with (audio or voice or speech) with text and(datbase or	USOCR; FPRS; EPO; JPO; DERWENT;	OR	ON	2017/07/20 13:10

		data adj2 base or librar$4) same (tag$4 or metatag$4 or meta$3) and (quer$3 or search$3 or retriv$6)	IBM_TDB			

EAST Search History (Interference)

<This search history is empty>

7/ 20/ 2017 1:17:47 PM
C:\ Users\ isherali\ Documents\ EAST\ Workspaces\ ish.wsp

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	2928	(captur$5 or record$3) with ((audio or speech or voice) with image) and (convert$3 or conversion) with (audio or voice or speech)and(datbase or data adj2 base or librar$4 archiv$6)	USPAT	OR	ON	2017/07/20 12:57
L2	722	(captur$5 or record$3) with ((audio or speech or voice) with image) and (convert$3 or conversion) with (audio or voice or speech) with text and(datbase or data adj2 base or librar$4 archiv$6)	USPAT	OR	ON	2017/07/20 12:57
L3	524	(captur$5 or record$3) with ((audio or speech or voice) with image) and (convert$3 or conversion) with (audio or voice or speech) with text and(datbase or data adj2 base or librar$4) and search$3	USPAT	OR	ON	2017/07/20 12:59
L4	165	(captur$5 or record$3) with ((audio or speech or voice) with image) and (convert$3 or conversion) with (audio or voice or speech) with text and(datbase or data adj2 base or librar$4) same (tag$4 or metatag$6 or meta$6) and search$3	USPAT	OR	ON	2017/07/20 13:01
L5	3	(captur$5 or record$3) with ((audio or speech or voice) with image) same camera and (convert$3 or conversion) with (audio or voice or speech) with text and(datbase or data adj2 base or librar$4) same (tag$4 or metatag$6 or meta$6) same search$3	USPAT	OR	ON	2017/07/20 13:03
L6	31	(captur$5 or record$3) with ((audio or speech) with image) same camera and (convert$3 or conversion) with (audio or voice or speech) with text and(datbase or data adj2 base or librar$4) same (tag$4 or metatag$6 or meta$6) and search$3	USPAT	OR	ON	2017/07/20 13:05
L7	89	(captur$5 or record$3) with ((audio or speech) with image) same camera and (convert$3 or conversion) with (audio or voice or speech) with text and(datbase or data adj2 base or librar$4) same (tag$4 or metatag$6 or meta$6) and search$3	US-PGPUB; USPAT	OR	ON	2017/07/20 13:07
L8	89	(captur$5 or record$3) with ((audio or speech) with image) same camera and (convert$3 or conversion) with (audio or voice or speech) with text and(datbase or data adj2 base or librar$4) same (tag$4 or metatag$6 or meta$6) and (quer$3 or search$3 or retriv$6)	US-PGPUB; USPAT	OR	ON	2017/07/20 13:08
L9	0	(captur$5 or record$3) with ((audio or speech) with image) same camera and (convert$3 or conversion) with (audio or voice or speech) with text and(datbase or	USOCR; FPRS; EPO; JPO; DERWENT;	OR	ON	2017/07/20 13:10

		data adj2 base or librar$4) same (tag$4 or metatag$4 or meta$3) and (quer$3 or search$3 or retriv$6)	IBM_TDB			

EAST Search History (Interference)

<This search history is empty>

7/ 20/ 2017 1:11:54 PM
C:\ Users\ isherali\ Documents\ EAST\ Workspaces\ ish.wsp

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	15272013	MALONE, MICHAEL F.
	Examiner	**Art Unit**
	ISHRAT I SHERALI	2667

CPC						
Symbol					**Type**	**Version**
H04L		9	/	14	F	2013-01-01
G06F		17	/	30038	I	2013-01-01
G06F		17	/	30244	I	2013-01-01
G06F		17	/	30749	I	2013-01-01
H04L		9	/	3263	I	2013-01-01
H04L		9	/	3271	I	2013-01-01
H04L		9	/	3297	I	2013-01-01
H04L		2209	/	60	A	2013-01-01
H04L		2209	/	80	A	2013-01-01
H04N		5	/	76	I	2013-01-01
G06F		17	/	30268	I	2013-01-01
G06F		17	/	3028	I	2013-01-01
H04N		5	/	44	I	2013-01-01
H04N		21	/	25816	I	2013-01-01
H04N		21	/	25875	I	2013-01-01
H04N		21	/	26603	I	2013-01-01
H04N		21	/	26613	I	2013-01-01
H04N		21	/	2747	I	2013-01-01
H04N		21	/	4108	I	2013-01-01
H04N		21	/	6582	I	2013-01-01
H04N		21	/	835	I	2013-01-01
H04N		21	/	84	I	2013-01-01
H04N		5	/	77	A	2013-01-01
H04N		5	/	9201	A	2013-01-01

CPC Combination Sets							
Symbol					**Type**	**Set**	**Ranking**	**Version**
			/					
			/					

NONE		**Total Claims Allowed:**	
(Assistant Examiner)	(Date)	17	
/ISHRAT I SHERALI/ Primary Examiner.Art Unit 2667	7/20/17	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	4

U.S. Patent and Trademark Office — Part of Paper No. 20170720

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	15272013	MALONE, MICHAEL F.
	Examiner	**Art Unit**
	ISHRAT I SHERALI	2667

NONE (Assistant Examiner)	(Date)	**Total Claims Allowed:** 17	
/ISHRAT I SHERALI/ Primary Examiner.Art Unit 2667 (Primary Examiner)	7/20/17 (Date)	O.G. Print Claim(s) 1	O.G. Print Figure 4

U.S. Patent and Trademark Office

Part of Paper No. 20170720

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	15272013	MALONE, MICHAEL F.
	Examiner	**Art Unit**
	ISHRAT I SHERALI	2667

US ORIGINAL CLASSIFICATION		INTERNATIONAL CLASSIFICATION										
CLASS	SUBCLASS	CLAIMED					NON-CLAIMED					
382	100	H	0	4	L	9 / 14 (2006.0)						
CROSS REFERENCE(S)		G	0	6	F	17 / 30 (2006.0)						
CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)											
382	305											

NONE (Assistant Examiner)	(Date)	**Total Claims Allowed:** 17	
/ISHRAT I SHERALI/ Primary Examiner.Art Unit 2667 (Primary Examiner)	7/20/17 (Date)	O.G. Print Claim(s) 1	O.G. Print Figure 4

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	15272013	MALONE, MICHAEL F.
	Examiner	**Art Unit**
	ISHRAT I SHERALI	2667

☒ Claims renumbered in the same order as presented by applicant ☐ CPA ☒ T.D. ☐ R.1.47

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
	1		17	13	33										
	2		18	14	34										
	3		19	15	35										
	4		20	16	36										
	5	1	21	17	37										
	6	2	22												
	7	3	23												
	8	4	24												
	9	5	25												
	10	6	26												
	11	7	27												
	12	8	28												
	13	9	29												
	14	10	30												
	15	11	31												
	16	12	32												

NONE (Assistant Examiner)	(Date)	**Total Claims Allowed:** 17	
/ISHRAT I SHERALI/ Primary Examiner.Art Unit 2667	7/20/17	O.G. Print Claim(s) 1	O.G. Print Figure 4
(Primary Examiner)	(Date)		

Doc Code: DIST.E.FILE **Document Description: Electronic Terminal Disclaimer - Filed**	PTO/SB/26 U.S. Patent and Trademark Office Department of Commerce

Electronic Petition Request	**TERMINAL DISCLAIMER TO OBVIATE A DOUBLE PATENTING REJECTION OVER A "PRIOR" PATENT**
Application Number	15272013
Filing Date	21-Sep-2016
First Named Inventor	MICHAEL MALONE
Attorney Docket Number	MPOR-33317
Title of Invention	APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/VIDEO IMAGE/ OBJECT RECOGNITION, CREATION OF SEARCHABLE MATATAG(S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL

☒ Filing of terminal disclaimer does not obviate requirement for response under 37 CFR 1.111 to outstanding Office Action

☒ This electronic Terminal Disclaimer is not being used for a Joint Research Agreement.

Owner	Percent Interest
MYPORT IP, INC.	100%

The owner(s) with percent interest listed above in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term of prior patent number(s)

7778438

as the term of said prior patent is presently shortened by any terminal disclaimer. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and the prior patent are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.

In making the above disclaimer, the owner does not disclaim the terminal part of the term of any patent granted on the instant application that would extend to the expiration date of the full statutory term of the prior patent, "as the term of said prior patent is presently shortened by any terminal disclaimer," in the event that said prior patent later:
- expires for failure to pay a maintenance fee;
- is held unenforceable;
- is found invalid by a court of competent jurisdiction;
- is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321;
- has all claims canceled by a reexamination certificate;
- is reissued; or
- is in any manner terminated prior to the expiration of its full statutory term as presently shortened by any terminal disclaimer.

◉ Terminal disclaimer fee under 37 CFR 1.20(d) is included with Electronic Terminal Disclaimer request.

○ I certify, in accordance with 37 CFR 1.4(d)(4), that the terminal disclaimer fee under 37 CFR 1.20(d) required for this terminal disclaimer has already been paid in the above-identified application.	
Applicant claims the following fee status: ○ Small Entity ○ Micro Entity ◉ Regular Undiscounted I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.	
THIS PORTION MUST BE COMPLETED BY THE SIGNATORY OR SIGNATORIES I certify, in accordance with 37 CFR 1.4(d)(4) that I am: ◉ An attorney or agent registered to practice before the Patent and Trademark Office who is of record in this application Registration Number 74472 ○ A sole inventor ○ A joint inventor; I certify that I am authorized to sign this submission on behalf of all of the inventors as evidenced by the power of attorney in the application ○ A joint inventor; all of whom are signing this request	
Signature	/Keith D. Harden, Reg. #74472/
Name	Keith D. Harden

*Statement under 37 CFR 3.73(b) is required if terminal disclaimer is signed by the assignee (owner).
Form PTO/SB/96 may be used for making this certification. See MPEP § 324.

Electronic Patent Application Fee Transmittal

Application Number:	15272013
Filing Date:	21-Sep-2016
Title of Invention:	APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/ VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE MATATAG (S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL
First Named Inventor/Applicant Name:	MICHAEL F. MALONE
Filer:	Keith D. Harden
Attorney Docket Number:	MPOR-33317

Filed as Large Entity

Filing Fees for Utility under 35 USC 111(a)

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
STATUTORY OR TERMINAL DISCLAIMER	1814	1	160	160
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Extension-of-Time:				
Miscellaneous:				
		Total in USD ($)		**160**

Doc Code: DISQ.E.FILE
Document Description: Electronic Terminal Disclaimer – Approved

Application No.: 15272013

Filing Date: 21-Sep-2016

Applicant/Patent under Reexamination: MALONE

Electronic Terminal Disclaimer filed on July 19, 2017

☒ APPROVED

This patent is subject to a terminal disclaimer

☐ DISAPPROVED

Approved/Disapproved by: Electronic Terminal Disclaimer automatically approved by EFS-Web

U.S. Patent and Trademark Office

Electronic Acknowledgement Receipt	
EFS ID:	29828715
Application Number:	15272013
International Application Number:	
Confirmation Number:	8747
Title of Invention:	APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO/ VIDEO IMAGE/OBJECT RECOGNITION, CREATION OF SEARCHABLE MATATAG (S)/ CONTEXTUAL TAG(S), STORAGE AND SEARCH RETRIEVAL
First Named Inventor/Applicant Name:	MICHAEL F. MALONE
Customer Number:	25883
Filer:	Keith D. Harden
Filer Authorized By:	
Attorney Docket Number:	MPOR-33317
Receipt Date:	19-JUL-2017
Filing Date:	21-SEP-2016
Time Stamp:	14:31:14
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	DA
Payment was successfully received in RAM	$160
RAM confirmation Number	072017INTEFSW00000834200780
Deposit Account	200780
Authorized User	Keith Harden
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows: 37 CFR 1.17 (Patent application and reexamination processing fees) 37 CFR 1.21 (Miscellaneous fees and charges)	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Terminal Disclaimer-Filed (Electronic)	eTerminal-Disclaimer.pdf	33766 d9aed3dd832e77342efc89b0fe3753324ed90c4e	no	2
Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	30694 ccfcddbb92f22febc647d38389437d6dfd8657f7	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	64460		

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.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	MALONE, Michael F.
Serial No.:	15/272,013
Confirmation No.:	8747
Filed:	September 21, 2016
Group:	2667
Examiner:	SHERALI, Ishrat I.
For:	METHOD FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH CONVERSION, METATAGS CREATION, STORAGE AND SEARCH RETRIEVAL

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

Dear Sir:

AMENDMENT AND RESPONSE TO OFFICE ACTION

This communication is responsive to the Examiner's Office Action dated March 24, 2017. Reconsideration is respectfully requested.

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

Amendments to the Claims are reflected in the listing of claims beginning on page 3 of this paper.

Remarks/Arguments begin on page 8 of this paper.

In the Specification

Please replace the title with the following amended title:

~~METHOD~~ APPARATUS FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH TO TEXT CONVERSION, PHOTO / VIDEO IMAGE / OBJECT RECOGNITION, CREATION OF SEARCHABLE METATAG(S) / CONTEXTUAL TAG(S), ~~METATAGS CREATION,~~ STORAGE AND SEARCH RETRIEVAL

In the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

1-20. (Canceled)

21. (New) A system for capturing image and audio information for storage in a database at a location on a network, comprising:

a microphone interfacable with an external audio information source that generates external audio information and a first data converter for capturing the first external audio information from the microphone,

a camera interfacing with an external image source to capture an image therefrom;

the first data converter processing the captured external audio information and storing it in a first digital audio format as stored digital audio within the capture device, the camera for processing the captured image and storing it as a stored digital image;

a transmitter for transmitting the stored digital audio and stored digital image to the location on the network; and

a system disposed at the location on the network and including:

a receiver for receiving the transmitted digital audio and digital image from the transmitter as a received set of captured information,

a system data converter for converting the received digital audio to a text based searchable file as a text context tag and creating an image recognition searchable context tag with image recognition of at least a portion of the digital image and associating the text and image recognition context tags with the digital image, and

the database storing the digital image in association with the text and image recognition context tags.

22. (New) The system of Claim 21, wherein the first data converter captures the first external audio information from the microphone during generation thereof.

23. (New) The system of Claim 21, wherein the camera captures the image from the external image source at an instant in time.

24. (New) The system of Claim 21, wherein the first data converter processes the captured external audio information from a start event representing the time that capture of the external audio information is initiated to a stop event representing the time that capture of the external audio information is complete.

25. (New) The system of Claim 24, wherein the transmitter transmits the stored digital audio information and the stored digital image to the location on the network after at least the stop event associated with the processing of the captured external audio information.

26. (New) A system for capturing image and audio information for storage, comprising:

a capture device having:

internal storage;

a microphone interfacable with and external audio information source that generates external audio information and a first data converter for capturing the first external audio information from the microphone,

a camera interfacing with and external image source to capture an image therefrom;

the first data converter processing the captured external audio information and storing it in a first digital audio format as stored digital audio in internal storage within the capture device, the camera for processing the captured image and storing it as a stored digital image in internal storage and a combiner for generating an association between the stored digital audio and the stored digital image,

a media data converter for converting the received set of captured information to convert the received digital audio to a text based searchable file as a text context tag and creating an image recognition searchable context tag with image recognition of at least a portion of the digital image and associating the text and image recognition context tags with the digital image, and

the internal storage storing the digital image in association with the text and image recognition context tags.

27. (New) The system of Claim 26, wherein the first data converter captures the first external audio information from the microphone during generation thereof.

28. (New) The system of Claim 26, wherein the camera captures the image from the external image source at an instant in time.

29. (New) The system of Claim 26, wherein the first data converter processes the captured external audio information from a start event representing the time that capture of the external audio information is initiated to a stop event representing the time that capture of the external audio information is complete.

30. (New) The system of Claim 26, and further including:

a transmitter associated with the capture device for transmitting the associated stored digital image in association with the text and image recognition context tags to the location on the network;

a system disposed at the location on a network and including:

a receiver for receiving the transmitted associated stored digital image in association with the text and image recognition context tags from the transmitter associated with the capture device as a received set of captured information,

a database, and

the database storing the received associated stored digital image in association with the text and image recognition context tags.

31. (New) The system of Claim 30, wherein the first data converter processes the captured external audio information from a start event representing the time that capture of the external audio information is initiated to a stop event representing the time that capture of the external audio information is complete.

32. (New) The system of Claim 31, wherein the transmitter transmits the associated stored digital image in association with the text and image recognition context tags to the location on the network after at least the stop event associated with the processing of the captured external audio information.

33. (New) A system for capturing image and audio information for storage, comprising:

internal storage;

a microphone interfacable with an external audio information source that generates external audio information and a first data converter for capturing the first external audio information from the microphone;

a camera interfacing with an image source to capture an image therefrom;

the first data converter processing the captured external audio information and storing it in a first digital audio format as stored digital audio within the capture device, the camera for processing the captured image and storing it as a stored digital image;

a second data converter for converting the received digital audio to a text based searchable file as a text context tag and creating an image recognition searchable context tag with image recognition of at least a portion of the digital image and associating the text and image recognition context tags with the digital image; and

the internal storage storing the digital image in association with the text and image recognition context tags.

34. (New) The system of Claim 33, wherein the image source is an external image source.

35. (New) The system of Claim 33, wherein the first data converter captures the first external audio information from the microphone during generation thereof.

36. (New) The system of Claim 33, wherein the camera captures the image from the image source at an instant in time.

37. (New) The system of Claim 33, wherein the first data converter processes the captured external audio information from a start event representing the time that capture of the external

audio information is initiated to a stop event representing the time that capture of the external audio information is complete.

REMARKS

Applicant has carefully reviewed the Office Action dated March 24, 2017. Applicant has canceled Claims 1-20 and added new Claims 21-37 to more clearly point out the present inventive concept. Reconsideration and favorable action is respectfully requested.

Double Patenting Rejections

Claims 1-20 stand rejected under 35 U.S.C. § 101 as claiming the same invention as that of claims 1-20 of prior U.S. Patent No. 8,509,477. Claims 1-20 also stand rejected on the ground of non-statutory obviousness-type double patenting as being unpatentable over claims 1-12 of U.S. Patent No. 7,778,438.

As noted above, Applicant has canceled Claims 1-20 and added new Claims 21-37. Applicant respectfully submits that new Claims 21-37 add limitations that are patentably distinct from original Claims 1-20. Therefore, Applicant respectfully requests withdrawal of the double patenting rejections.

Applicant has now made an earnest attempt in order to place this case in condition for allowance. For the reasons stated above, Applicant respectfully requests full allowance of the claims as amended. Please charge any additional fees or deficiencies in fees or credit any overpayment to Deposit Account No. 20-0780/MPOR-33317 of HOWISON & ARNOTT, L.L.P.

Respectfully submitted,
HOWISON & ARNOTT, L.L.P.
Attorneys for Applicant

/Keith D. Harden, Reg. #74472/
Keith D. Harden
Registration No. 74,472

KDH/ps

P.O. Box 741715
Dallas, Texas 75374-1715
Tel: 972-479-0462
Fax: 972-479-0464
June 26, 2017

Electronic Acknowledgement Receipt	
EFS ID:	29613153
Application Number:	15272013
International Application Number:	
Confirmation Number:	8747
Title of Invention:	METHOD FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH CONVERSION, METATAGS CREATION, STORAGE AND SEARCH RETRIEVAL
First Named Inventor/Applicant Name:	MICHAEL F. MALONE
Customer Number:	25883
Filer:	Keith D. Harden
Filer Authorized By:	
Attorney Docket Number:	MPOR-33317
Receipt Date:	26-JUN-2017
Filing Date:	21-SEP-2016
Time Stamp:	19:41:50
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		MPO33317AMD.pdf	100160 7529b64ac51e58fdefeeba7753529cd1ac40192c	yes	9

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Amendment/Req. Reconsideration-After Non-Final Reject	1	1
	Specification	2	2
	Claims	3	7
	Applicant Arguments/Remarks Made in an Amendment	8	9
Warnings:			
Information:			
	Total Files Size (in bytes):	100160	

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.

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 15/272,013	Filing Date 09/21/2016	☐ 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 $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =	
☐ 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

AMENDMENT	06/26/2017	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 17	Minus	** 20	= 0	X $80 =	0
	Independent (37 CFR 1.16(h))	* 3	Minus	***3	= 0	X $420 =	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**

AMENDMENT		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =	
	☐ 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.
** 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 number found in the appropriate box in column 1.

LIE
JASON B. EADDY

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.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
15/272,013	09/21/2016	MICHAEL F. MALONE	MPOR-33317	8747

25883 7590 03/24/2017

HOWISON & ARNOTT, L.L.P
P.O. BOX 741715
DALLAS, TX 75374-1715

EXAMINER	
SHERALI, ISHRAT I	

ART UNIT	PAPER NUMBER
2667	

NOTIFICATION DATE	DELIVERY MODE
03/24/2017	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

patents@dalpat.com

Office Action Summary	Application No. 15/272,013	Applicant(s) MALONE, MICHAEL F.	
	Examiner ISHRAT I. SHERALI	Art Unit 2667	AIA (First Inventor to File) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

5)☒ Claim(s) *1-20* is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6)☐ Claim(s) _____ is/are allowed.

7)☒ Claim(s) *1-20* is/are rejected.

8)☐ Claim(s) _____ is/are objected to.

9)☐ Claim(s) _____ are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a)☐ All b)☐ Some** c)☐ None of the:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☒ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b) Paper No(s)/Mail Date _____.

3) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

4) ☐ Other: _____.

Notice of Pre-AIA or AIA Status

1. The present application is being examined under the pre-AIA first to invent provisions.

Double Patenting

2. A rejection based on double patenting of the "same invention" type finds its support in the language of 35 U.S.C. 101 which states that "whoever invents or discovers any new and useful process... may obtain a patent therefor..." (Emphasis added). Thus, the term "same invention," in this context, means an invention drawn to identical subject matter. See *Miller v. Eagle Mfg. Co.*, 151 U.S. 186 (1894); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Ockert*, 245 F.2d 467, 114 USPQ 330 (CCPA 1957).

A statutory type (35 U.S.C. 101) double patenting rejection can be overcome by canceling or amending the claims that are directed to the same invention so they are no longer coextensive in scope. The filing of a terminal disclaimer cannot overcome a double patenting rejection based upon 35 U.S.C. 101.

3. Claims 1-20 rejected under 35 U.S.C. 101 as claiming the same invention as that of claim 1-20 of prior U.S. Patent No. 8,509,477. This is a statutory double patenting rejection.

The filing of a terminal disclaimer cannot overcome a double patenting rejection based upon 35 U.S.C. 101.

4. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on nonstatutory double patenting provided the reference application or patent either is shown to be commonly owned with the examined application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement. See MPEP § 717.02 for applications subject to examination under the first inventor to file provisions of the AIA as explained in MPEP § 2159. See MPEP §§ 706.02(l)(1) - 706.02(l)(3) for applications not subject to examination under the first inventor to file provisions of the AIA. A terminal disclaimer must be signed in compliance with 37 CFR 1.321(b).

The USPTO Internet website contains terminal disclaimer forms which may be used. Please visit www.uspto.gov/patent/patents-forms. The filing date of the application in which the form is filed determines what form (e.g., PTO/SB/25, PTO/SB/26, PTO/AIA/25, or PTO/AIA/26) should be used. A web-based eTerminal Disclaimer may be filled out completely online using web-screens. An eTerminal Disclaimer that meets all requirements is auto-processed and approved immediately upon submission. For more information about eTerminal Disclaimers, refer to www.uspto.gov/patents/process/file/efs/guidance/eTD-info-I.jsp.

5. Claims 1-20 are rejected on the ground of non-statutory obviousness-type double patenting as being un-patentable over claims 1-12 of U.S. Patent No. 7,778,438. Although the conflicting claims are not identical, they are not patentably distinct from each other because they claiming the same subject matter of converting information in another format and storing in the database and also independent claim 1 of the U.S Patent No.7,778,438 includes all of the limitations recited in independent claims 1 of the instant application, except claim 1 of the instant application also include "transmitting defined set of first captured information and the defined set of stored captured information to a remote location . Since independent claim 1 of the U.S Patent No 7,778,438 recites in the preamble "comprising" i.e. claim 1 of the U.S Patent No 7,778,438 is open- ended claim therefore it also include transmitting defined set of first captured information and the defined set of stored captured information to a remote location and furthermore independent claim 1 of the U.S Patent No 7,778,438 also

recite captured information and remote database. Similarly claims 2-20 of the instant application are similarly obvious over claims 1-12 of the U.S. Patent No. 7,778,358.

Communication

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sherali Ishrat whose telephone number is 571-272-7398. The examiner can normally be reached on 8:00 AM - 4:30PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matthew Bella can be reached on 571-272-7453. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/ISHRAT I SHERALI/
Primary Examiner, Art Unit 2667

November 12, 2015

Notice of References Cited	Application/Control No. 15/272,013	Applicant(s)/Patent Under Reexamination MALONE, MICHAEL F.	
	Examiner ISHRAT I. SHERALI	Art Unit 2667	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	CPC Classification	US Classification
*	A	US-8,983,119 B2	03-2015	Malone; Michael F.	G06F17/30038	382/100
*	B	US-8,509,477 B2	08-2013	Malone; Michael F.	G06F17/30038	382/100
*	C	US-7,778,438 B2	08-2010	Malone; Michael F.	G06F17/30038	382/100
*	D	US-7,043,048 B1	05-2006	Ellingson; Eric E.	H04N1/32101	382/100
*	E	US-6,990,444 B2	01-2006	Hind; John R.	G10L21/06	704/235
*	F	US-9,471,592 B2	10-2016	Koyama; Toshiaki	H04L12/1831	1/1
	G	US-				
	H	US-				
	I	US-				
	J	US-				
	K	US-				
	L	US-				
	M	US-				

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	CPC Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	21	(captur$5 or record$3) with (audio or speech or voice) same conver$8 same (datbase or data adj2 base or librar$4 or archiv$6) same (verif$6 or authenticat$3)	USPAT	OR	ON	2015/11/15 19:25
L2	42	(captur$5 or record$3) with (audio or speech or voice) same (convert$3 or conversion) same (datbase or data adj2 base or librar$4 or archiv$6) same (verif$6 or authenticat$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/11/15 19:27
L3	222	(captur$5 or record$3) with (audio or speech or voice) same (convert$3 or conversion) same (datbase or data adj2 base or librar$4 or archiv$6) and ((verif$6 or authenticat$3) with user)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/11/15 19:28
L4	121	(captur$5 or record$3) with (audio or speech or voice) same (convert$3 or conversion) same (datbase or data adj2 base or librar$4 or archiv$6) and ((verif$6 or authenticat$3) with user) same (receiv$3 or transmit$6)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/11/15 19:29
L9	1	(captur$5 or record$3) with (audio or speech or voice) same conver$8 same (datbase or data adj2 base or librar$4 or archiv$6) same (verif$6 or authenticat$3) same hash$3	USPAT	OR	ON	2015/11/15 19:35

EAST Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L5	54	(captur$5 or record$3) with (audio or speech or voice) same (convert$3 or conversion) same (datbase or data adj2 base or librar$4 or archiv$6) and ((verif$6 or authenticat$3) with user) same (receiv$3 or transmit$6)	USPAT	OR	ON	2015/11/15 19:29
L6	121	(captur$5 or record$3) with (audio or speech or voice) same (convert$3 or conversion) same (datbase or data adj2 base or librar$4 or archiv$6) and ((verif$6 or authenticat$3) with user) same (receiv$3 or transmit$6)	US-PGPUB; USPAT	OR	ON	2015/11/15 19:30
L7	16	(captur$5 or record$3) with (audio or speech or voice) same (convert$3 or conversion) same (datbase or data adj2 base or librar$4 or archiv$6) and ((verif$6 or authenticat$3) with user) same (receiv$3 or transmit$6) same hash$3	US-PGPUB; USPAT	OR	ON	2015/11/15 19:30
L8	16	7 AND ((G06F17/30244 OR G06F17/30268 OR	US-	OR	ON	2015/11/15

		G06F17/3028 OR G06F21/10 OR G06F2211/007 OR G06F2221/0708 OR G06F2221/2107 OR H04L2209/60 OR H04L2209/80 OR H04L9/3263 OR H04L9/3297 OR H04L65/4084 OR H04L67/306 OR H04L63/0428 OR H04L63/061 OR H04L2463/101 OR H04L2463/102 OR H04L67/18).CPC.)	PGPUB; USPAT			19:33

11/15/2015 7:38:28 PM

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	15272013	MALONE, MICHAEL F.
	Examiner	Art Unit
	ISHRAT I SHERALI	2667

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	03/20/2017								
	1	✓								
	2	✓								
	3	✓								
	4	✓								
	5	✓								
	6	✓								
	7	✓								
	8	✓								
	9	✓								
	10	✓								
	11	✓								
	12	✓								
	13	✓								
	14	✓								
	15	✓								
	16	✓								
	17	✓								
	18	✓								
	19	✓								
	20	✓								

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	15272013
Filing Date	2016-09-21
First Named Inventor	MALONE, Michael F.
Art Unit	2161
Examiner Name	
Attorney Docket Number	MPOR-33317

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	5267042		1993-11-30	TSUCHIYA ET AL.	
	2	6947571		2005-09-20	RHOADS ET AL.	
	3	6807534		2004-10-19	ERICKSON	
	4	6788800		2004-09-07	CARR ET AL	
	5	6786397		2004-09-07	SILVERBROOK 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					

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

FOREIGN PATENT DOCUMENTS Remove

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2]	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							

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		

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

EXAMINER SIGNATURE

Examiner Signature	/ISHRAT I SHERALI/	Date Considered	03/19/2017

*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.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

Doc code: IDS
Doc description: Information Disclosure Statement (IDS) Filed

PTO/SB/08a (03-15)
Approved for use through 07/31/2016. 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	15272013
Filing Date	2016-09-21
First Named Inventor	MALONE, Michael F.
Art Unit	2161
Examiner Name	
Attorney Docket Number	MPOR-33317

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	8424227		2013-04-23	HARRINGTON, JOHN	
	2	8099772		2012-01-17	TAKADA ET AL.	
	3	6968453		2005-11-22	DOYLE ET AL.	
	4	6996251		2006-02-07	MALONE ET AL.	
	5	4742369		1988-05-03	ISHII ET AL.	
	6	5031122		1991-07-09	WITTY	
	7	5276472		1994-01-04	BELL ET AL.	
	8	5546145		1996-08-13	BERNARDI ET AL.	

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	9	5786851		1998-07-28	KONDO ET AL.	
	10	5940121		1999-08-17	MCINTYRE ET AL.	
	11	5995936		1999-11-30	BRAIS ET AL.	
	12	6031526		2000-02-29	SHIPP	
	13	7617542		2009-11-10	VATAJA	
	14	6104430		2000-08-15	FUKUOKA	
	15	6683649		2004-01-27	ANDERSON	
	16	7053938		2006-05-30	SHERRY	
	17	7907199		2011-03-15	SEKI ET AL.	
	18	7831598		2010-11-09	KO	
	19	7324943		2008-01-29	RIGAZIO ET AL.	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	20	7961218		2011-06-14	SEKI ET AL.	
	21	6115137		2000-09-05	OZAWA ET AL	
	22	6111605		2000-08-09	SUZUKI	
	23	7319484		2008-01-15	YOSHIDA ET AL	
	24	6222985		2001-04-24	MIYAKE	
	25	6741864		2004-05-25	WILCOCK, ET AL.	
	26	6636158		2003-10-21	BANDO, ET AL.	
	27	6525768		2003-02-25	OBRADOVICH	
	28	6720879		2004-04-03	EDWARDS	
	29	6507371		2003-01-14	HASHIMOTO, ET AL.	
	30	6498586		2002-12-24	PANKINAHO	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	31	6292633		2001-09-18	NAKAGAWA	
	32	6198989		2001-03-06	TANKHILEVICH, ET AL.	
	33	5768640		1998-06-16	TAKAHASHI, ET AL.	
	34	5642285		1997-06-24	WOO, ET AL.	
	35	6560660		2003-05-06	FLANAGIN	
	36	6526351		2003-02-25	WHITHAM	
	37	6160964		2000-12-12	IMOTO	
	38	6532298		2003-03-11	CAMBIER, ET AL.	
	39	6205249		2001-03-20	MOSKOWITZ	
	40	6185683		2001-02-06	GINTER, ET AL.	
	41	6714778		2004-03-30	NYKANEN, ET AL.	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	42	6166729		2000-12-26	ACOSTA, EDWARD	
	43	6424968		2002-07-23	BROSTER, ET AL.	
	44	6065119		2000-05-16	SANDFORD, II, ET AL.	
	45	6462778		2002-10-08	ABRAM, ET AL.	
	46	6505160		2003-01-07	LEVY, ET AL.	
	47	6526215		2003-02-25	HIRAI, ET AL.	
	48	6606117		2003-08-12	WINDLE	
	49	2950971		1960-08-30	LEWIN	
	50	3439598		1969-04-22	WEITZNER, ET AL.	
	51	5602458		1997-02-11	DOWE	
	52	5692225		1997-11-25	BERNARDI, ET AL.	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	53	6681029		2004-01-20	RHOADS	
	54	7010144		2006-03-07	DAVIS ET AL.	
	55	7043048		2006-05-09	ELLINGSON	
	56	0055422		1866-06-05	ROUSTAEI	
	57	4015240		1977-03-29	SWONGER ET AL.	
	58	4109237		1978-08-22	HILL	
	59	4528588		1985-07-09	LOFBERG	
	60	4620318		1986-10-28	HILL	
	61	4994831		1991-02-19	MARANDI	
	62	4995086		1991-02-19	LILLEY ET AL.	
	63	5103486		1992-04-07	GRIPPI	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

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

Application Number	15272013
Filing Date	2016-09-21
First Named Inventor	MALONE, Michael F.
Art Unit	2161
Examiner Name	
Attorney Docket Number	MPOR-33317

	64	5767496		1998-06-16	SWARTZ ET AL.	
	65	5787183		1998-07-28	SCHROEDER	
	66	5799092		1998-08-25	KRISTOL ET AL.	
	67	5841886		1998-11-24	RHOADS	
	68	5857038		1999-01-05	OWADA ET AL.	
	69	5907149		1999-05-25	MARCKINI	
	70	5995630		1999-11-30	BORZA	
	71	6005936		1999-12-21	SCHIMIZU ET AL.	
	72	6064764		2000-05-16	BHASKARAN ET AL.	
	73	6148091		2000-11-24	DIMARIA	
	74	6185316		2001-02-06	BUFFAM	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	75	6208746		2001-03-27	MUSGRAVE	
	76	6243481		2001-06-05	TAO	
	77	6292092		2001-09-18	CHOW ET AL.	
	78	6301368		2001-10-09	BOLLE ET AL.	
	79	6321981		2001-11-27	RAY ET AL.	
	80	6366680		2002-04-02	BRUNK ET AL.	
	81	6377699		2002-04-23	MUSGRAVE ET AL.	
	82	6389151		2002-05-14	CARR ET AL.	
	83	6397334		2002-05-28	CHAINER ET AL.	
	84	6577336		2003-06-10	SAFAI	
	85	6687383		2004-02-03	KANEVSKY ET AL.	

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	86	7778440		2010-08-17	MALONE, MICHAEL F.	
	87	8068638		2011-11-29	MALONE, MICHAEL F.	
	88	7184573		2007-02-27	MALONE ET AL.	
	89	6023241		2000-02-08	CLAPPER	
	90	5913078		1999-06-15	KIMURA ET AL.	
	91	5732354		1998-03-24	MACDONALD	
	92	6212401		2001-04-03	ACKLEY	
	93	5825890		1998-10-20	ELGAMAL ET AL.	
	94	8135169		2012-03-13	MALONE, MICHAEL F.	
	95	6914695		2005-07-05	WALTERS ET AL.	
	96	5247300		1993-09-21	SOHN, SUNG W.	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	97	7778438		2010-08-17	MALONE, MICHAEL F.	
	98	RE39526		2007-03-20	HULL ET AL.	
	99	52670425		1993-11-30	TSUCHIYA ET AL.	
	100	5515042		1996-05-07	NELSON	
	101	5712679		1998-01-27	COLES	
	102	5726660		1998-03-10	PURDY ET AL.	
	103	5806005		1998-09-08	HULL ET AL.	
	104	6181373		2001-01-30	COLES	
	105	6359837		2002-03-19	TSUKAMOTO	
	106	6571271		2003-05-27	SAVITZKY ET AL.	
	107	6831682		2004-12-14	SILVERBROOK ET AL.	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	108	6833861		2004-12-21	MATSUMOTO ET AL.	
	109	7016899		2006-03-21	STERN ET AL.	
	110	7265779		2007-09-04	SATO ET AL.	
	111	6889324		2005-05-03	KANAI ET AL.	
	112	6853987		2005-02-08	COOK, DAVID P.	
	113	6269446		2001-07-31	SCHUMACHER ET AL.	
	114	6968366		2005-11-22	ZHANG ET AL.	
	115	7391960		2008-06-24	SHINOZAKI ET AL.	
	116	6990444		2006-01-24	HIND ET AL.	
	117	5737491		1998-04-07	ALLEN ET AL.	
	118	4115805		1978-09-19	MORTON	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	119	4334241		1982-06-08	KASHIOKA, ET AL.	
	120	4574319		1986-03-04	KONISHI	
	121	5335072		1994-08-02	TANAKA, ET AL.	
	122	5506644		1996-04-09	SUZUKI, ET AL.	
	123	5633678		1997-05-27	PARULSKI, ET AL.	
	124	6332193		2001-12-18	GLASS, ET AL.	
	125	6219560		2001-04-17	ERKKILA, MARKO	
	126	6014183		2000-01-11	HOANG	
	127	6115717		2000-09-05	MEHROTRA ET AL.	
	128	5819289		1998-10-06	SANFORD, II ET AL.	
	129	5657077		1997-08-12	DEANGELIS, ET AL.	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

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

Application Number	15272013
Filing Date	2016-09-21
First Named Inventor	MALONE, Michael F.
Art Unit	2161
Examiner Name	
Attorney Docket Number	MPOR-33317

	130	5682458		1997-10-28	FUNAZAKI	
	131	5692104		1997-11-25	CHOW, ET AL.	
	132	5706457		1998-01-06	DWYER, ET AL.	
	133	5796428		1998-08-18	MATSUMOTO, ET AL.	
	134	5828809		1998-10-27	CHANG, ET AL.	
	135	5835667		1998-11-10	WACTLAR, ET AL.	
	136	5872865		1999-02-16	NORMILE, ET AL.	
	137	5889578		1999-03-30	JAMZADEH	
	138	5893095		1999-04-06	JAIN, ET AL.	
	139	6282362		2001-08-28	MURPHY, ET AL.	
	140	5499294		1996-03-12	FRIEDMAN, GARY L.	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	141	5815201		1998-09-29	HASHIMOTO ET AL.	
	142	4389109		1983-06-21	TANIGUCHI ET AL.	
	143	4270853		1981-06-02	HATADA ET AL.	
	144	4270854		1981-06-02	STEMME ET AL.	
	145	4344682		1982-08-17	HATTORI	
	146	4443077		1984-04-17	TANIKAWA	
	147	4613911		1986-09-23	OHTA	
	148	4642717		1987-02-10	MATSUDA ET AL.	
	149	4905029		1990-02-27	KELLEY	
	150	4951079		1990-08-21	HOSHINO ET AL.	
	151	4965626		1990-10-23	ROBISON ET AL.	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	152	4977419		1990-12-11	WASH ET AL.	
	153	4983996		1991-01-08	KINOSHITA	
	154	5023635		1991-06-11	NEALON	
	155	5025283		1991-06-18	ROBINSON	
	156	5027149		1991-06-25	HOSHINO ET AL.	
	157	5070355		1991-12-03	INOUE ET AL.	
	158	5097278		1992-03-17	TAMAMURA ET AL.	
	159	5099262		1992-03-24	TANAKA ET AL.	
	160	5128700		1992-07-07	INOUE ET AL.	
	161	5142310		1992-08-25	TANIGUCHI ET AL.	
	162	5146249		1992-09-08	HODA ET AL.	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	163	5160952		1992-11-03	IWASHITA ET AL.	
	164	5245372		1993-09-14	OASHIMA	
	165	5313235		1994-05-17	INOUE ET AL.	
	166	5426745		1995-06-20	BAJI ET AL.	
	167	6952164		2005-10-04	JUNQUA	
	168	6959868		2005-11-01	TSIKOS ET AL.	
	169	6965324		2005-11-15	SUGGS, SR.	
	170	6956671		2005-10-18	MONTY ET AL.	
	171	6895126		2005-05-17	DI BERNARDO ET AL.	
	172	6856344		2005-02-15	FRANZ	
	173	6750902		2004-06-15	STEINBERG ET AL.	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

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

Application Number	15272013
Filing Date	2016-09-21
First Named Inventor	MALONE, Michael F.
Art Unit	2161
Examiner Name	
Attorney Docket Number	MPOR-33317

	174	6747692		2004-06-08	PATEL ET AL.	
	175	6670984		2003-12-30	TANAKA ET AL.	
	176	6628325		2003-12-30	STEINBERG ET AL.	
	177	6535298		2003-03-18	WINTER ET AL	
	178	6411328		2002-06-25	FRANKE ET AL.	
	179	7095871		2006-08-22	JONES ET AL.	
	180	5841978		1998-11-24	RHOADS	
	181	5410598		1995-04-25	SHEAR	
	182	5493677		1996-02-20	BALOGH ET AL.	
	183	5502576		1996-03-26	RAMSAY ET AL.	
	184	5519778		1996-05-21	LEIGHTON ET AL.	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	185	5530759		1996-06-25	BRAUDAWAY ET AL.	
	186	5568570		1996-10-22	RABBANI	
	187	5581800		1996-12-03	FARDEAU ET AL.	
	188	5617119		1997-04-01	BRIGGS ET AL.	
	189	5629980		1997-05-13	STEFIK	
	190	5646990		1997-07-08	LI	
	191	5646997		1997-07-08	BARTSON	
	192	5740244		1998-04-14	INDECK ET AL.	
	193	5765152		1998-06-09	ERICKSON	
	194	5822432		1998-10-13	MOSKOWITZ ET AL.	
	195	5845281		1998-12-01	BENSON	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

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

Application Number	15272013
Filing Date	2016-09-21
First Named Inventor	MALONE, Michael F.
Art Unit	2161
Examiner Name	
Attorney Docket Number	MPOR-33317

	196	5862260		1999-01-19	RHOADS	
	197	5892900		1999-04-06	GINTER ET AL.	
	198	5923327		1999-07-13	SMITH ET AL.	
	199	5943422		1999-08-24	VAN WIE ET AL.	
	200	5978773		1999-11-02	HUDETZ ET AL.	
	201	5991876		1999-11-23	JOHNSON ET AL.	
	202	6014569		2000-01-11	BOTTUM	
	203	6122403		2000-09-19	RHOADS	
	204	6141753		2000-10-31	ZHAO ET AL.	
	205	6233684		2001-05-15	STEFIK ET AL.	
	206	6243480		2001-06-05	ZHAO ET AL.	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	207	6282654		2001-08-28	IKEDA ET AL.	
	208	6300880		2001-10-09	SITNIK	
	209	6334187		2001-12-25	KADONO	
	210	6363043		2002-03-26	KONDO	
	211	6389538		2002-05-14	GRUSE ET AL.	
	212	6421450		2002-07-16	NAKANO	
	213	6437933		2002-08-20	SUGIYAMA ET AL.	
	214	6449367		2002-09-10	VAN WIE ET AL.	
	215	6469969		2002-10-22	CARSON ET AL.	
	216	6491217		2002-12-10	CATAN	
	217	6560339		2003-05-06	IWAMURA	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	218	6591365		2003-07-08	COOKSON	
	219	6611607		2003-08-26	DAVIS ET AL	
	220	6636249		2003-10-21	REKIMOTO	

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	20040114042		2004-06-17	PAOLINI, MICHAEL A., ET AL.	
	2	20030177094		2003-09-18	NEEDHAM, BRADFORD H., ET AL.	
	3	20040103283		2004-05-27	HOMAK, ZOLTAN	
	4	20040101138		2004-05-27	REVITAL, DAN, ET AL.	
	5	20040091111		2004-05-13	LEVY, KENNETH L., ET AL.	
	6	20040073568		2004-04-15	YONAHA, MAKOTO	

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	7	20040073557		2004-04-15	PICCIONELLI, GREGORY A., ET AL.	
	8	20040068371		2004-04-08	ESTEP, RANDALL S.	
	9	20040053637		2004-03-18	IIDA, TAKAYUKI	
	10	20040049734		2004-03-11	SIMSKE, STEVEN J.	
	11	20040044911		2004-03-04	TAKADA, MASAYUKI	
	12	20040039930		2004-02-26	OHMORI, MOTOJI, ET AL.	
	13	20040032499		2004-02-19	SILVERBROOK, KIA, ET AL.	
	14	20040023686		2004-02-05	KING, JOHN J., ET AL.	
	15	20040005078		2004-01-08	TILLOTSON, SCOTT ANDREW	
	16	20040022444		2004-02-05	RHOADS, GEOFFREY B.	
	17	20020073056		2002-06-13	BROSTER, IAN, ET AL.	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

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

Application Number	15272013
Filing Date	2016-09-21
First Named Inventor	MALONE, Michael F.
Art Unit	2161
Examiner Name	
Attorney Docket Number	MPOR-33317

	18	20010012062		2001-08-09	ANDERSON	
	19	20020191087		2002-12-19	HASHIMOTO, ET AL.	
	20	20030048921		2003-03-13	CAHALL, NATHAN D.	
	21	20020194480		2002-12-19	NAGAO, KATASHI	
	22	20030011684		2003-01-16	NARAYANASWAMI ET AL.	
	23	20030032033		2003-02-13	ANGLIN ET AL.	
	24	20050043018		2005-02-24	KAWAMOTO, YOJI	
	25	20060018506		2006-01-26	RODRIGUEZ ET AL.	
	26	20070055689		2007-03-08	RHOADS ET AL.	
	27	20010007130		2001-07-05	TAKARAGI	
	28	20010011680		2001-08-09	SOLTESZ ET AL.	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	29	20010025342		2001-09-27	UCHIDA	
	30	20010034835		2001-10-25	SMITH	
	31	20020010684		2002-01-24	MOSKOWITZ	
	32	20020061120		2002-05-23	CARR ET AL.	
	33	20020072935		2002-06-13	ROWSE ET AL.	
	34	20010015759		2001-08-23	SQUIBBS	
	35	20010012066		2001-08-09	PARULSKI ET AL.	
	36	20010018349		2001-08-30	KINNUNEN ET AL.	
	37	20020046188		2002-04-18	BURGES ET AL.	
	38	20040197013		2004-10-07	KAMEI	
	39	20050134703		2005-06-23	MITTAL	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	40	20070250526		2007-10-25	HANNA	
	41	20070098172		2007-05-03	LEVY ET AL.	
	42	20030187950		2003-10-02	RISING III	
	43	20050231752		2005-10-20	SAINIO, KARI	
	44	20050219359		2005-10-06	TRELA, RICHARD STEVEN	
	45	20050216949		2005-09-29	CANDELORA ET AL.	
	46	20050216580		2005-09-29	RAJI ET AL.	
	47	20050185049		2005-08-25	IWAI ET AL.	
	48	20050143136		2005-06-30	LEV ET AL.	
	49	20050094000		2005-05-05	SON ET AL.	
	50	20050091311		2005-04-28	LUND ET AL.	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	51	20050041035		2005-02-24	NAGATOMO ET AL.	
	52	20050024493		2005-02-03	NAM, KI Y.	
	53	20040257431		2004-12-23	GIRISH ET AL.	
	54	20040203386		2004-10-14	TISCHER ET AL.	
	55	20040183915		2004-09-23	GOTOHDA, ET AL.	
	56	20040148518		2004-07-29	GRUNDBACK ET AL.	
	57	20040109199		2004-06-10	TSUBAKI, HISAYOSHI	
	58	20040085203		2004-05-06	JUNQUA, JEAN-CLAUDE	
	59	20040070670		2004-04-15	FOSTER, WARD S.	
	60	20010021144		2001-09-13	OSHIMA ET AL.	
	61	20010021978		2001-09-13	OKAYASU ET AL.	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

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

Application Number	15272013
Filing Date	2016-09-21
First Named Inventor	MALONE, Michael F.
Art Unit	2161
Examiner Name	
Attorney Docket Number	MPOR-33317

	62	20010044824		2001-11-22	HUNTER ET AL.	
	63	20020001395		2002-01-03	DAVIS ET AL.	
	64	20020015042		2002-02-07	ROBOTHAM ET AL.	
	65	20020030907		2002-03-14	IKEDA ET AL.	
	66	20020031240		2002-03-14	LEVY ET AL.	
	67	20020032502		2002-03-14	RUSSELL, J. CHRIS	
	68	20020033844		2002-03-21	LEVY ET AL.	
	69	20020051577		2002-05-02	KINJO, NAOTO	
	70	20020059162		2002-05-16	SHINODA ET AL.	
	71	20020062382		2002-05-23	RHOADS ET AL.	
	72	20020075298		2002-06-20	SCHENA ET AL.	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	73	20020080271		2002-06-27	EVELEENS ET AL.	
	74	20020080396		2002-06-27	SILVERBROOK ET AL.	
	75	20020080964		2002-06-27	STONE ET AL.	
	76	20020083123		2002-06-27	FREEDMAN ET AL.	
	77	20020102966		2002-08-01	LEV ET AL.	
	78	20020146148		2002-10-10	LEVY, KENNETH L.	
	79	20020152388		2002-10-17	LINNARTZ 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	0905966	EP		1999-03-31	CANON KABUSHIKI KAISHA		
	2	0152178	WO		2001-07-19	DIGIMARC CORPORATION		

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	3	9901859	WO		1999-01-14	IMIGE INCORPORATED		
	4	0167707	WO		2001-09-13	SCALE EIGHT, INCORPORATED		
	5	1204277	EP		2002-05-08	TEKTRONIX, INC.		
	6	9743736	WO		1997-11-20	DIGIMARC CORPORATION		
	7	9626494	WO		1996-08-29	THE REGENTS OF THE UNIVERSITY OF CALIFORNIA		
	8	9936876	WO		1999-07-22	DIGIMARC CORPORATION		
	9	0115021	WO		2001-03-01	DIGITAL CONVERGENCE.:COM INC.		
	10	0161508	WO		2001-08-23	DIGIMARC CORORATION		
	11	0173586	WO		2001-10-04	DIGITAL: CONVERGENCE CORPORATION		
	12	0175629	WO		2001-10-11	NEOMEDIA TECHNOLOGIES, INC.		
	13	0197128	WO		2001-12-20	MARKANY INC.		

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	14	0211446	WO		2002-02-07	KONINKLIJKE PHILIPS ELECTRONICS, N.V.		
	15	9827510	WO		1998-06-25	THE REGENTS OF THE UNIVERSITY OF CALIFORNIA		
	16	0036605	WO		2000-06-22	KENT RIDGE DIGITAL LABS		
	17	0070585	WO		2000-11-23	DIGIMARC CORPORATION		
	18	9603286	WO		1996-02-08	INTERNATIONAL DATA MATRIX, INC.		

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	United States Patent and Trademark Office; Request for Ex Parte Reexamination in related re-exam application 90/013,265; dated June 6, 2014; Examiner Angela M. Lie; 1085 pages 6 Jun 2014	
	2	United States Patent and Trademark Office; Request for Ex Parte Reexamination in related re-exam application 90/013,255; dated May 27, 2014; Examiner Scott Louis Weaver; 906 pages 27 May 2014	
	3	United States Patent and Trademark Office; Request for Ex Parte Reexamination in related re-exam application 90/013,265; dated February 16, 2015; Examiner Angela M. Lie; 187 pages 16 Feb 2015	
	4	United States Patent and Trademark Office; Request for Ex Parte Reexamination in related re-exam application 90/013,255; dated March 2, 2015; Examiner Angela M. Lie; 177 pages 2 Mar 2015	

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	5	H. KRAWCZYK, RFC 2104 (RFC2104) RFC 2104 - HMAC: Keyed-Hashing for Message Authentication https://www.faqs.org/rfcs/rfc2104.htm, February 1997.	
	6	G. Friedman, "The Trustworthy Digital Camera: Restoring Credibility To The Photographic Image," IEEE Transactions on Consumer Electronics, pages 905-910, Vol. 39, No. 4, November 1993.	
	7	FRANKS, et al., "HTTP Authentication: Basic and Digest Access Authentication," The Internet Society, June 1999.	
	8	FRANKS, et al., "An Extension to HTTP: Digest Access Authentication," The Internet Society, January 1997.	
	9	Supplemental First Amended Complaint, MyPort IP, Inc. v. HTC Corporation, Civil Action No. 6:11-cv-00246 (E.D. Tex. Filed March 8, 2012) ("MyPort Complaint"), 12 pages. 8 Mar 2012	
	10	Digital Still Camera Image File Format Standard (Exchangeable image file format for Digital Still Cameras: Exif), Version 2.1, published June 12, 1998 by the Japan Electronic Industry Development Association (JEIDA) ("Exif 2.1 Specification"), 173 pages.	
	11	TIFF Specification Revision 6.0, Aldus Developers Desk, Aldus Corporation, June 3, 1992.	
	12	James Bao-Yen Tsui, Fundamental of Global Positioning System Receivers: A Software Approach, pages 1-6, 73-108, 1983-217 (John Wiley & Sons, Inc. 2000).	
	13	PPP Encryption Control Prorocol, Internet Engineering Task Force (IETF), published June 1996.	
	14	Alfred Menezes, Paul C. van Oorschot, and Scott A. Vanstone, Handbook of Applied Cryptography (CRC Press 1996).	
	15	MyPort IP, Inc.'s Disclosure Pursuant to Patent Rule 3-1, MyPort IP, Inc. v. HTC Corporation, Civil Action No. 6:11-cv-00246 (E.D. Tex. Filed May 13, 2011).	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

	16	STANSELL, Jr., "Civil GPS from a Future Perspective," Proceedings of the IEEE, October 1983, vol. 71, No. 10; pgs. 1187-1192. 1 Oct 1983	
	17	DALE, DIANA & ROG, RON, The Need For a Meta-Tag Standard for Audio and Visual Materials, Proc. Int. Conf. on Dublin Core and Metadata for e-Communities 2002: 205-206. 1 Jan 2002	
	18	KOSOVIC, DOUGLAS; SCHROETER, RONALD; and HUNTER, JANE, Collaborative Video Annotation, Indexing and Discussion over high-bandwidth networks, DSTC Technical Report TR247, 2002, DSTC Pty Ltd., St Lucia, Qld, Australia. 1 Jan 2002	

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

EXAMINER SIGNATURE

Examiner Signature	/ISHRAT I SHERALI/	Date Considered	03/19/2017

*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.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /I.I.S/

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	15272013	MALONE, MICHAEL F.
	Examiner	Art Unit
	ISHRAT I SHERALI	2667

CPC- SEARCHED		
Symbol	Date	Examiner

CPC COMBINATION SETS - SEARCHED		
Symbol	Date	Examiner

US CLASSIFICATION SEARCHED			
Class	Subclass	Date	Examiner
382	100, 190, 305	3/19/17	IS
705	26.1-27.1	3/19/17	IS

SEARCH NOTES		
Search Notes	Date	Examiner
SEARCH EAST TEXT LIMITED CPC/USPC, INVENTOR/ASSINGEE, GOOGLE	3/19/17	IS

INTERFERENCE SEARCH			
US Class/ CPC Symbol	US Subclass / CPC Group	Date	Examiner

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

BIB DATA SHEET

CONFIRMATION NO. 8747

SERIAL NUMBER	FILING or 371(c) DATE RULE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
15/272,013	09/21/2016	345	2667	MPOR-33317

APPLICANTS
MYPORT TECHNOLOGIES, INC., McKinney, TX;

INVENTORS
MICHAEL F. MALONE, McKinney, TX;

**** CONTINUING DATA ***************************
This application is a CON of 14/660,166 03/17/2015 ABN
which is a CON of 13/965,625 08/13/2013 PAT 8983119
which is a CON of 13/417,229 03/10/2012 PAT 8509477
which is a CON of 12/857,358 08/16/2010 PAT 8135169
which is a CON of 11/621,062 01/08/2007 PAT 7778438
which claims benefit of 60/757,075 01/06/2006
and is a CIP of 11/325,373 01/04/2006 PAT 7184573
which is a CON of 10/674,910 09/29/2003 PAT 6996251
which claims benefit of 60/414,449 09/30/2002

**** FOREIGN APPLICATIONS ***************************

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ****
10/03/2016

Foreign Priority claimed ☐ Yes ☑ No 35 USC 119(a-d) conditions met ☐ Yes ☑ No Verified and Acknowledged /ISHRAT I SHERALI/ Examiner's Signature	☐ Met after Allowance Initials	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
		TX	4	20	2

ADDRESS
HOWISON & ARNOTT, L.L.P
P.O. BOX 741715
DALLAS, TX 75374-1715
UNITED STATES

TITLE
METHOD FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH CONVERSION, METATAGS CREATION, STORAGE AND SEARCH RETRIEVAL

FILING FEE RECEIVED		
1600	FEES: Authority has been given in Paper No._____ to charge/credit DEPOSIT ACCOUNT No._____ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other _____ ☐ Credit

BIB (Rev. 05/07).

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
15/272,013	09/21/2016	MICHAEL F. MALONE	MPOR-33317

CONFIRMATION NO. 8747

25883
HOWISON & ARNOTT, L.L.P
P.O. BOX 741715
DALLAS, TX 75374-1715

PUBLICATION NOTICE

OC000000088465354

Title:METHOD FOR PERSONAL VOICE ASSISTANT, LOCATION SERVICES, MULTI-MEDIA CAPTURE, TRANSMISSION, SPEECH CONVERSION, METATAGS CREATION, STORAGE AND SEARCH RETRIEVAL

Publication No.US-2017-0011106-A1
Publication Date:01/12/2017

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 Office of Public Records. The Office of Public Records 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, Office of Public Records, 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 http://pair.uspto.gov/. 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

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	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

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	5267042		1993-11-30	TSUCHIYA ET AL.	
	2	6947571		2005-09-20	RHOADS ET AL.	
	3	6807534		2004-10-19	ERICKSON	
	4	6788800		2004-09-07	CARR ET AL	
	5	6786397		2004-09-07	SILVERBROOK 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					
If you wish to add additional U.S. Published Application citation information please click the Add button.						Add

FOREIGN PATENT DOCUMENTS	Remove

EFS Web 2.1.17

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2]	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							

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		

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.

[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	15272013
	Filing Date	2016-09-21
	First Named Inventor	MALONE, Michael F.
	Art Unit	2161
	Examiner Name	
	Attorney Docket Number	MPOR-33317

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	/Keith D. Harden, Reg. #74472/	Date (YYYY-MM-DD)	2017-01-04
Name/Print	Keith D. Harden	Registration Number	74472

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 : G06K 9/36 — A1

(11) International Publication Number: WO 97/43736

(43) International Publication Date: 20 November 1997 (20.11.97)

(21) International Application Number: PCT/US97/08351

(22) International Filing Date: 16 May 1997 (16.05.97)

(30) Priority Data:

08/649,419	16 May 1996 (16.05.96)	US
08/746,613	12 November 1996 (12.11.96)	US

(60) Parent Applications or Grants
(63) Related by Continuation
US 08/649,419 (CIP)
Filed on 16 May 1996 (16.05.96)
US 08/746,613 (CIP)
Filed on 12 November 1996 (12.11.96)

(71) Applicant *(for all designated States except US)*: DIGIMARC CORPORATION [US/US]; 521 S.W. 11th Avenue #200, Portland, OR 97205 (US).

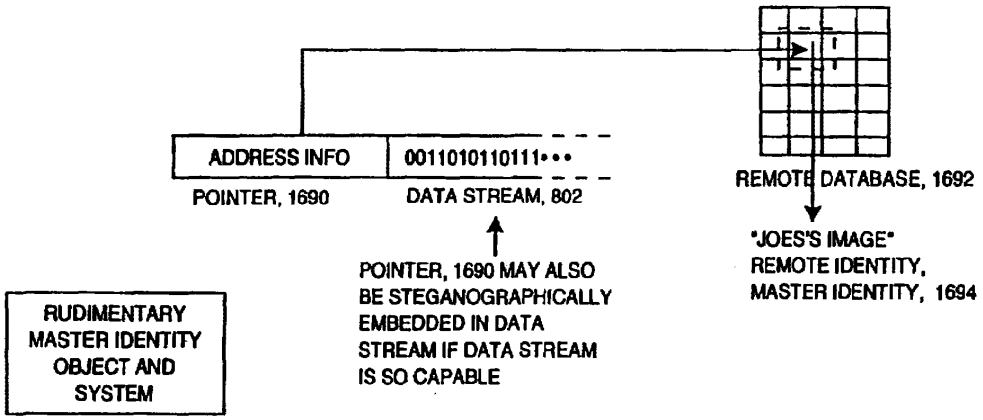
(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: RHOADS, Geoffrey, B. [US/US]; 304 S.W. Tualatin Loop, West Linn, OR 97068 (US).

(74) Agents: JOHNSON, Alexander, C., Jr. et al.; Marger, Johnson, McCollom & Stolowitz, 1030 S.W. Morrison Street, Portland, OR 97205 (US).

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

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: COMPUTER SYSTEM LINKED BY USING INFORMATION IN DATA OBJECTS



(57) Abstract

Various improvements to steganographic systems, and applications therefore, are disclosed. The improvements include facilitating scale and rotation registration for steganographic decoding by use of rotationally symmetric steganographically embedded patterns and subliminal digital graticules; improved techniques for decoding without access to unencoded originals; improving robustness of steganographic coding in motion pictures and/or in the presence of lossy compression/decompression; and representing data (1690) by patterned bit cells (802) whose energy in the spatial domain facilitates decoding registration (1692). Applications include enhanced-security financial transactions, counterfeit resistant identification cards, fraud deterrent systems for cellular telephony, covert modem channels in video transmissions, photo duplication kiosks with automatic copyright detection, and hotlinked image objects (e.g., with embedded URLs) for use on the Internet.

FOR THE PURPOSES OF INFORMATION ONLY

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

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

COMPUTER SYSTEM LINKED BY USING INFORMATION IN DATA OBJECTS

Background of the Invention

Hiding data in imagery or audio is a technique well known to artisans in the field, and is termed "steganography." There are a number of diverse approaches to, and applications of, steganography. A brief survey follows:

British patent publication 2,196,167 to Thorn EMI discloses a system in which an audio recording is electronically mixed with a marking signal indicative of the owner of the recording, where the combination is perceptually identical to the original. U.S. patents 4,963,998 and 5,079,648 disclose variants of this system.

U.S. Patent 5,319,735 to Bolt, Berenak & Newman rests on the same principles as the earlier Thorn EMI publication, but additionally addresses psycho-acoustic masking issues.

U.S. Patents 4,425,642, 4,425,661, 5,404,377 and 5,473,631 to Moses disclose various systems for imperceptibly embedding data into audio signals -- the latter two patents particularly focusing on neural network implementations and perceptual coding details.

U.S. Patent 4,943,973 to AT&T discloses a system employing spread spectrum techniques for adding a low level noise signal to other data to convey auxiliary data therewith. The patent is particularly illustrated in the context of transmitting network control signals along with digitized voice signals.

U.S. Patent 5,161,210 to U.S. Philips discloses a system in which additional low-level quantization levels are defined on an audio signal to convey, e.g., a copy inhibit code, therewith.

U.S. Patent 4,972,471 to Gross discloses a system intended to assist in the automated monitoring of audio (e.g. radio) signals for copyrighted materials by reference to identification signals subliminally embedded therein.

U.S. Patent 5,243,423 to DeJean discloses a video steganography system which encodes digital data (e.g. program syndication verification, copyright marking, media research, closed captioning, or like data) onto randomly selected video lines. DeJean relies

on television sync pulses to trigger a stored pseudo random sequence which is XORed with the digital data and combined with the video.

European application EP 581,317 discloses a system for redundantly marking images with multi-bit identification codes. Each "1" ("0") bit of the code is manifested as a slight increase (decrease) in pixel values around a plurality of spaced apart "signature points." Decoding proceeds by computing a difference between a suspect image and the original, unencoded image, and checking for pixel perturbations around the signature points.

PCT application WO 95/14289 describes the present applicant's prior work in this field.

Komatsu et al., describe an image marking technique in their paper "A Proposal on Digital Watermark in Document Image Communication and Its Application to Realizing a Signature," Electronics and Communications in Japan, Part 1, Vol. 73, No. 5, 1990, pp. 22-33. The work is somewhat difficult to follow but apparently results in a simple yes/no determination of whether the watermark is present in a suspect image (e.g. a 1 bit encoded message).

Others embed the information "in-band" (i.e. in the visible video signal itself). Examples include U.S. Patents 4,528,588, 4,595,950, and 5,319,453; European application 441,702; and Matsui et. al, "Video-Steganography: How to Secretly Embed a Signature in a Picture," IMA Intellectual Property Project Proceedings, January 1994, Vol. 1, Issue 1, pp. 187-205.

Copyright marking of video and multimedia is found in "Access Control and Copyright Protection for Images (ACCOPI), WorkPackage 8: Watermarking," June 30, 1995, 46 pages. TALISMAN, appears to extend certain of the ACCOPI work.

A British company, Highwater FBI, Ltd., has introduced a software product which is said to imperceptibly embed identifying information into photographs and other graphical images. This technology is the subject of European patent applications 9400971.9 (filed January 19, 1994), 9504221.2 (filed March 2, 1995), and 9513790.7 (filed July 3, 1995), the first of which has been laid open as PCT publication WO 95/20291.

Brief Description Of The Drawings

Fig. 1 is a simple and classic depiction of a one dimensional digital signal which is discretized in both axes.

Fig. 2 is a general overview, with detailed description of steps, of the process of embedding an "imperceptible" identification signal onto another signal.

Fig. 3 is a step-wise description of how a suspected copy of an original is identified.

Fig. 4 is a schematic view of an apparatus for pre-exposing film with identification information.

Fig. 5 is a diagram of a "black box" embodiment.

Fig. 6 is a schematic block diagram of the embodiment of Fig. 5.

Fig. 7 shows a variant of the Fig. 6 embodiment adapted to encode successive sets of input data with different code words but with the same noise data.

Fig. 8 shows a variant of the Fig. 6 embodiment adapted to encode each frame of a videotaped production with a unique code number.

Figs. 9A-9C are representations of an industry standard noise second.

Fig. 10 shows an integrated circuit used in detecting standard noise codes.

Fig. 11 shows a process flow for detecting a standard noise code that can be used in the Fig. 10 embodiment.

Fig. 12 is an embodiment employing a plurality of detectors.

Fig. 13 shows an embodiment in which a pseudo-random noise frame is generated from an image.

Fig. 14 illustrates how statistics of a signal can be used in aid of decoding.

Fig. 15 shows how a signature signal can be preprocessed to increase its robustness in view of anticipated distortion, e.g. MPEG.

Figs. 16 and 17 show embodiments in which information about a file is detailed both in a header, and in the file itself.

Figs. 18-20 show details of embodiments using rotationally symmetric patterns.

Fig. 21 shows encoding "bumps" rather than pixels.

Figs. 22-26 detail aspects of a security card.

Fig. 27 is a diagram illustrating a network linking method using information embedded in data objects that have inherent noise.

Figs. 27A and 27B show a typical web page, and a step in its encapsulation into a self extracting web page object.

Fig. 28 is a diagram of a photographic identification document or security card.

Figs. 29 and 30 illustrate two embodiments by which subliminal digital graticules can be realized.

Fig. 29A shows a variation on the Fig. 29 embodiment.

Figs. 31A and 31B show the phase of spatial frequencies along two inclined axes.

Figs. 32A - 32C show the phase of spatial frequencies along first, second and third concentric rings.

Figs 33A - 33E show steps in the registration process for a subliminal graticule using inclined axes.

Figs. 34A - 34E show steps in the registration process for a subliminal graticule using concentric rings.

Figs. 35A - 35C shows further steps in the registration process for a subliminal graticule using inclined axes.

Figs. 36A - 36D show another registration process that does not require a 2D FFT.

Fig. 37 is a flow chart summarizing a registration process for subliminal graticules.

Fig. 38 is a block diagram showing principal components of an exemplary wireless telephony system.

Fig. 39 is a block diagram of an exemplary steganographic encoder that can be used in the telephone of the Fig. 38 system.

Fig. 40 is a block diagram of an exemplary steganographic decoder that can be used in the cell site of the Fig. 1 system.

Figs. 41A and 41B show exemplary bit cells used in one form of encoding.

Fig. 42 shows a hierarchical arrangement of signature blocks, sub-blocks, and bit cells used in one embodiment.

Fig. 43 shows a general overview of a computer system linked by using information embedded in the objects.

Figs. 44-59 are representative computer screens generated in accordance with the present invention.

Fig. 60 is a basic overview of how a traditional file header can be replaced by a pointer to a database location where header information resides.

Fig. 61 expands on Fig. 60 by illustrating that a pointer to a remote location can coexist with existing data file formats which might include headers.

Fig. 62 is a basic map showing how an instrument or application can associate a pointer attached to an object to some remote digital storage location.

Fig. 63 is a condensed depiction of a process that an instrument or application might go through in resolving and acting upon remote information pointed to by an attached pointer.

Fig. 64 graphically illustrates that attached information can actually be sequences of actions or multiple sets of information.

Fig. 65 shows how the basic principles of master identity objects can be used to fuel a reporting network wherein a central location can "keep track" of where its copies migrate.

Detailed Description

In the following discussion of an illustrative embodiment, the words "signal" and "image" are used interchangeably to refer to both one, two, and even beyond two dimensions of digital signal. Examples will routinely switch back and forth between a one dimensional audio-type digital signal and a two dimensional image-type digital signal.

In order to fully describe the details of an illustrative embodiment, it is necessary first to describe the basic properties of a digital signal. Fig. 1 shows a classic representation of a one dimensional digital signal. The x-axis defines the index numbers of sequence of digital "samples," and the y-axis is the instantaneous value of the signal at that sample, being constrained to exist only at a finite number of levels defined as the "binary depth" of a digital sample. The example depicted in Fig. 1 has the value of 2 to the fourth power, or "4 bits," giving 16 allowed states of the sample value.

For audio information such as sound waves, it is commonly accepted that the digitization process discretizes a continuous phenomena both in the time domain and in the signal level domain. As such, the process of digitization itself introduces a fundamental error source, in that it cannot record detail smaller than the discretization interval in either domain. The industry has referred to this, among other ways, as "aliasing" in the time domain, and "quantization noise" in the signal level domain. Thus, there will always be a basic error floor of a digital signal. Pure quantization noise, measured in a root mean square sense, is theoretically known to have the value of one over the square root of twelve, or about 0.29 DN, where DN stands for 'Digital Number' or the finest unit increment of the signal level. For example, a perfect 12-bit digitizer will have 4096 allowed DN with an innate root mean square noise floor of ~0.29 DN.

All known physical measurement processes add additional noise to the transformation of a continuous signal into the digital form. The quantization noise typically adds in quadrature (square root of the mean squares) to the "analog noise" of the measurement process, as it is sometimes referred to.

With almost all commercial and technical processes, the use of the decibel scale is used as a measure of signal and noise in a given recording medium. The expression "signal-to-noise ratio" is generally used, as it will be in this disclosure. As an example, this disclosure refers to signal to noise ratios in terms of signal power and noise power, thus 20 dB represents a 10 times increase in signal amplitude.

In summary, this embodiment embeds an N-bit value onto an entire signal through the addition of a very low amplitude encodation signal which has the look of pure noise. N is usually at least 8 and is capped on the higher end by ultimate signal-to-noise considerations and "bit error" in retrieving and decoding the N-bit value. As a practical matter, N is chosen based on application specific considerations, such as the number of unique different "signatures" that are desired. To illustrate, if N=128, then the number of unique digital signatures is in excess of 10^^38 (2^^128). This number is believed to be more than adequate to both identify the material with sufficient statistical certainty and to index exact sale and distribution information.

The amplitude or power of this added signal is determined by the aesthetic and informational considerations of each and every application using the present methodology. For instance, non-professional video can stand to have a higher embedded signal level without becoming noticeable to the average human eye, while high precision audio may only be able to accept a relatively small signal level lest the human ear perceive an objectionable increase in "hiss." These statements are generalities and each application has its own set of criteria in choosing the signal level of the embedded identification signal. The higher the level of embedded signal, the more corrupted a copy can be and still be identified. On the other hand, the higher the level of embedded signal, the more objectionable the perceived noise might be, potentially impacting the value of the distributed material.

To illustrate the range of different applications to which applicant's technology can be applied, the present specification details two different systems. The first (termed, for lack of a better name, a "batch encoding" system), applies identification coding to an existing data signal. The second (termed, for lack of a better name, a "real time encoding" system), applies identification coding to a signal as it is produced. Those skilled in the art will recognize that the principles of applicant's technology can be applied in a number of other contexts in addition to these particularly described.

The discussions of these two systems can be read in either order. Some readers may find the latter more intuitive than the former; for others the contrary may be true.

BATCH ENCODING

The following discussion of a first class of embodiments is best prefaced by a section defining relevant terms:

The original signal refers to either the original digital signal or the high quality digitized copy of a non-digital original.

The N-bit identification word refers to a unique identification binary value, typically having N range anywhere from 8 to 128, which is the identification code ultimately placed onto the original signal via the disclosed transformation process. In the illustrated embodiment, each N-bit identification word begins with the sequence of values '0101,' which is used to determine an optimization of the signal-to-noise ratio in the identification procedure of a suspect signal (see definition below).

The m'th bit value of the N-bit identification word is either a zero or one corresponding to the value of the m'th place, reading left to right, of the N-bit word. E.g., the first (m=1) bit value of the N=8 identification word 01110100 is the value '0;' the second bit value of this identification word is '1', etc.

The m'th individual embedded code signal refers to a signal which has dimensions and extent precisely equal to the original signal (e.g. both are a 512 by 512 digital image), and which is (in the illustrated embodiment) an independent pseudo-random sequence of digital values. "Pseudo" pays homage to the difficulty in philosophically defining pure randomness, and also indicates that there are various acceptable ways of generating the "random" signal. There will be exactly N individual embedded code signals associated with any given original signal.

The acceptable perceived noise level refers to an application-specific determination of how much "extra noise," i.e. amplitude of the composite embedded code signal described next, can be added to the original signal and still have an acceptable signal to sell or otherwise distribute. This disclosure uses a 1 dB increase in noise as a typical value which might be acceptable, but this is quite arbitrary.

The composite embedded code signal refers to the signal which has dimensions and extent precisely equal to the original signal, (e.g. both are a 512 by 512 digital image), and which contains the

addition and appropriate attenuation of the N individual embedded code signals. The individual embedded signals are generated on an arbitrary scale, whereas the amplitude of the composite signal must not exceed the pre-set acceptable perceived noise level, hence the need for "attenuation" of the N added individual code signals.

The distributable signal refers to the nearly similar copy of the original signal, consisting of the original signal plus the composite embedded code signal. This is the signal which is distributed to the outside community, having only slightly higher but acceptable "noise properties" than the original.

A suspect signal refers to a signal which has the general appearance of the original and distributed signal and whose potential identification match to the original is being questioned. The suspect signal is then analyzed to see if it matches the N-bit identification word.

The detailed methodology of this first embodiment begins by stating that the N-bit identification word is encoded onto the original signal by having each of the m bit values multiply their corresponding individual embedded code signals, the resultant being accumulated in the composite signal, the fully summed composite signal then being attenuated down to the acceptable perceived noise amplitude, and the resultant composite signal added to the original to become the distributable signal.

The original signal, the N-bit identification word, and all N individual embedded code signals are then stored away in a secured place. A suspect signal is then found. This signal may have undergone multiple copies, compressions and decompressions, resamplings onto different spaced digital signals, transfers from digital to analog back to digital media, or any combination of these items. IF the signal still appears similar to the original, i.e. its innate quality is not thoroughly destroyed by all of these transformations and noise additions, then depending on the signal to noise properties of the embedded signal, the identification process should function to some objective degree of statistical confidence. The extent of corruption of the suspect signal and the original acceptable perceived noise level are two key parameters in determining an expected confidence level of identification.

The identification process on the suspected signal begins by resampling and aligning the suspected signal onto the digital format and extent of the original signal. Thus, if an image has been reduced by a factor of two, it needs to be digitally enlarged by that same factor. Likewise, if a piece of music has been "cut out," but may still have the same sampling rate as the original, it is necessary to register this cut-out piece to the original, typically done by performing a local digital cross-correlation of the two signals (a common digital operation), finding at what delay value the correlation peaks, then using this found delay value to register the cut piece to a segment of the original.

Once the suspect signal has been sample-spacing matched and registered to the original, the signal levels of the suspect signal should be matched in an rms sense to the signal level of the original. This can be done via a search on the parameters of offset, amplification, and gamma being optimized by using the

minimum of the mean squared error between the two signals as a function of the three parameters. We can call the suspect signal normalized and registered at this point, or just normalized for convenience.

The newly matched pair then has the original signal subtracted from the normalized suspect signal to produce a difference signal. The difference signal is then cross-correlated with each of the N individual embedded code signals and the peak cross-correlation value recorded. The first four bit code ('0101') is used as a calibrator both on the mean values of the zero value and the one value, and on further registration of the two signals if a finer signal to noise ratio is desired (i.e., the optimal separation of the 0101 signal will indicate an optimal registration of the two signals and will also indicate the probable existence of the N-bit identification signal being present.)

The resulting peak cross-correlation values will form a noisy series of floating point numbers which can be transformed into 0's and 1's by their proximity to the mean values of 0 and 1 found by the 0101 calibration sequence. If the suspect signal has indeed been derived from the original, the identification number resulting from the above process will match the N-bit identification word of the original, bearing in mind either predicted or unknown "bit error" statistics. Signal-to-noise considerations will determine if there will be some kind of "bit error" in the identification process, leading to a form of X% probability of identification where X might be desired to be 99.9% or whatever. If the suspect copy is indeed not a copy of the original, an essentially random sequence of 0's and 1's will be produced, as well as an apparent lack of separation of the resultant values. This is to say, if the resultant values are plotted on a histogram, the existence of the N-bit identification signal will exhibit strong bi-level characteristics, whereas the non-existence of the code, or the existence of a different code of a different original, will exhibit a type of random gaussian-like distribution. This histogram separation alone should be sufficient for an identification, but it is even stronger proof of identification when an exact binary sequence can be objectively reproduced.

Specific Example

Imagine that we have taken a valuable picture of two heads of state at a cocktail party, pictures which are sure to earn some reasonable fee in the commercial market. We desire to sell this picture and ensure that it is not used in an unauthorized or uncompensated manner. This and the following steps are summarized in Fig. 2.

Assume the picture is transformed into a positive color print. We first scan this into a digitized form via a normal high quality black and white scanner with a typical photometric spectral response curve. (It is possible to get better ultimate signal to noise ratios by scanning in each of the three primary colors of the color image, but this nuance is not central to describing the basic process.)

Let us assume that the scanned image now becomes a 4000 by 4000 pixel monochrome digital image with a grey scale accuracy defined by 12-bit grey values or 4096 allowed levels. We will call this the "original digital image" realizing that this is the same as our "original signal" in the above definitions.

During the scanning process we have arbitrarily set absolute black to correspond to digital value '30'. We estimate that there is a basic 2 Digital Number root mean square noise existing on the original digital image, plus a theoretical noise (known in the industry as "shot noise") of the square root of the brightness value of any given pixel. In formula, we have:

$$<\text{RMS Noise}_{n,m}> = \text{sqrt}(4 + (V_{n,m}-30)) \quad (1)$$

Here, n and m are simple indexing values on rows and columns of the image ranging from 0 to 3999. Sqrt is the square root. V is the DN of a given indexed pixel on the original digital image. The < > brackets around the RMS noise merely indicates that this is an expected average value, where it is clear that each and every pixel will have a random error individually. Thus, for a pixel value having 1200 as a digital number or "brightness value", we find that its expected rms noise value is sqrt(1204) = 34.70, which is quite close to 34.64, the square root of 1200.

We furthermore realize that the square root of the innate brightness value of a pixel is not precisely what the eye perceives as a minimum objectionable noise, thus we come up with the formula:

$$<\text{RMS Addable Noise}_{n,m}> = X*\text{sqrt}(4+(V_{n,m}-30)\text{^}Y) \quad (2)$$

Where X and Y have been added as empirical parameters which we will adjust, and "addable" noise refers to our acceptable perceived noise level from the definitions above. We now intend to experiment with what exact value of X and Y we can choose, but we will do so at the same time that we are performing the next steps in the process.

The next step in our process is to choose N of our N-bit identification word.' We decide that a 16 bit main identification value with its 65536 possible values will be sufficiently large to identify the image as ours, and that we will be directly selling no more than 128 copies of the image which we wish to track, giving 7 bits plus an eighth bit for an odd/even adding of the first 7 bits (i.e. an error checking bit on the first seven). The total bits required now are at 4 bits for the 0101 calibration sequence, 16 for the main identification, 8 for the version, and we now throw in another 4 as a further error checking value on the first 28 bits, giving 32 bits as N. The final 4 bits can use one of many industry standard error checking methods to choose its four values.

We now randomly determine the 16 bit main identification number, finding for example, 1101 0001 1001 1110; our first versions of the original sold will have all 0's as the version identifier, and the

error checking bits will fall out where they may. We now have our unique 32 bit identification word which we will embed on the original digital image.

To do this, we generate 32 independent random 4000 by 4000 encoding images for each bit of our 32 bit identification word. The manner of generating these random images is revealing. There are numerous ways to generate these. By far the simplest is to turn up the gain on the same scanner that was used to scan in the original photograph, only this time placing a pure black image as the input, then scanning this 32 times. The only drawback to this technique is that it does require a large amount of memory and that "fixed pattern" noise will be part of each independent "noise image." But, the fixed pattern noise can be removed via normal "dark frame" subtraction techniques. Assume that we set the absolute black average value at digital number '100,' and that rather than finding a 2 DN rms noise as we did in the normal gain setting, we now find an rms noise of 10 DN about each and every pixel's mean value.

We next apply a mid-spatial-frequency bandpass filter (spatial convolution) to each and every independent random image, essentially removing the very high and the very low spatial frequencies from them. We remove the very low frequencies because simple real-world error sources like geometrical warping, splotches on scanners, mis-registrations, and the like will exhibit themselves most at lower frequencies also, and so we want to concentrate our identification signal at higher spatial frequencies in order to avoid these types of corruptions. Likewise, we remove the higher frequencies because multiple generation copies of a given image, as well as compression-decompression transformations, tend to wipe out higher frequencies anyway, so there is no point in placing too much identification signal into these frequencies if they will be the ones most prone to being attenuated. Therefore, our new filtered independent noise images will be dominated by mid-spatial frequencies. On a practical note, since we are using 12-bit values on our scanner and we have removed the DC value effectively and our new rms noise will be slightly less than 10 digital numbers, it is useful to boil this down to a 6-bit value ranging from -32 through 0 to 31 as the resultant random image.

Next we add all of the random images together which have a '1' in their corresponding bit value of the 32-bit identification word, accumulating the result in a 16-bit signed integer image. This is the unattenuated and un-scaled version of the composite embedded signal.

Next we experiment visually with adding the composite embedded signal to the original digital image, through varying the X and Y parameters of equation 2. In formula, we visually iterate to both maximize X and to find the appropriate Y in the following:

$$V_{dist;n,m} = V_{orig;n,m} + V_{comp;n,m} * X * \mathrm{sqrt}(4 + V_{orig;n,m} \hat{} Y) \qquad (3)$$

where dist refers to the candidate distributable image, i.e. we are visually iterating to find what X and Y will give us an acceptable image; orig refers to the pixel value of the original image; and comp refers to the pixel value of the composite image. The n's and m's still index rows and columns of the image and indicate that this operation is done on all 4000 by 4000 pixels. The symbol V is the DN of a given pixel and a given image.

As an arbitrary assumption, now, we assume that our visual experimentation has found that the value of X = 0.025 and Y=0.6 are acceptable values when comparing the original image with the candidate distributable image. This is to say, the distributable image with the "extra noise" is acceptably close to the original in an aesthetic sense. Note that since our individual random images had a random rms noise value around 10 DN, and that adding approximately 16 of these images together will increase the composite noise to around 40 DN, the X multiplication value of 0.025 will bring the added rms noise back to around 1 DN, or half the amplitude of our innate noise on the original. This is roughly a 1 dB gain in noise at the dark pixel values and correspondingly more at the brighter values modified by the Y value of 0.6.

So with these two values of X and Y, we now have constructed our first versions of a distributable copy of the original. Other versions will merely create a new composite signal and possibly change the X slightly if deemed necessary. We now lock up the original digital image along with the 32-bit identification word for each version, and the 32 independent random 4-bit images, waiting for our first case of a suspected piracy of our original. Storage wise, this is about 14 Megabytes for the original image and 32*0.5bytes*16 million = ~256 Megabytes for the random individual encoded images. This is quite acceptable for a single valuable image. Some storage economy can be gained by simple lossless compression.

Finding a Suspected Piracy of our Image

We sell our image and several months later find our two heads of state in the exact poses we sold them in, seemingly cut and lifted out of our image and placed into another stylized background scene. This new "suspect" image is being printed in 100,000 copies of a given magazine issue, let us say. We now go about determining if a portion of our original image has indeed been used in an unauthorized manner. Fig. 3 summarizes the details.

The first step is to take an issue of the magazine, cut out the page with the image on it, then carefully but not too carefully cut out the two figures from the background image using ordinary scissors. If possible, we will cut out only one connected piece rather than the two figures separately. We paste this onto a black background and scan this into a digital form. Next we electronically flag or mask out the black background, which is easy to do by visual inspection.

We now procure the original digital image from our secured place along with the 32-bit identification word and the 32 individual embedded images. We place the original digital image onto our computer screen using standard image manipulation software, and we roughly cut along the same borders as our masked area of the suspect image, masking this image at the same time in roughly the same manner. The word 'roughly' is used since an exact cutting is not needed, it merely aids the identification statistics to get it reasonably close.

Next we rescale the masked suspect image to roughly match the size of our masked original digital image, that is, we digitally scale up or down the suspect image and roughly overlay it on the original image. Once we have performed this rough registration, we then throw the two images into an automated scaling and registration program. The program performs a search on the three parameters of x position, y position, and spatial scale, with the figure of merit being the mean squared error between the two images given any given scale variable and x and y offset. This is a fairly standard image processing methodology. Typically this would be done using generally smooth interpolation techniques and done to sub-pixel accuracy. The search method can be one of many, where the simplex method is a typical one.

Once the optimal scaling and x-y position variables are found, next comes another search on optimizing the black level, brightness gain, and gamma of the two images. Again, the figure of merit to be used is mean squared error, and again the simplex or other search methodologies can be used to optimize the three variables. After these three variables are optimized, we apply their corrections to the suspect image and align it to exactly the pixel spacing and masking of the original digital image and its mask. We can now call this the standard mask.

The next step is to subtract the original digital image from the newly normalized suspect image only within the standard mask region. This new image is called the difference image.

Then we step through all 32 individual random embedded images, doing a local cross-correlation between the masked difference image and the masked individual embedded image. 'Local' refers to the idea that one need only start correlating over an offset region of +/- 1 pixels of offset between the nominal registration points of the two images found during the search procedures above. The peak correlation should be very close to the nominal registration point of 0,0 offset, and we can add the 3 by 3 correlation values together to give one grand correlation value for each of the 32 individual bits of our 32-bit identification word.

After doing this for all 32 bit places and their corresponding random images, we have a quasi-floating point sequence of 32 values. The first four values represent our calibration signal of 0101. We now take the mean of the first and third floating point value and call this floating point value '0,' and we take the mean of the second and the fourth value and call this floating point value '1.' We then step through all remaining 28 bit values and assign either a '0' or a '1' based simply on which mean value they are closer to. Stated simply, if the suspect image is indeed a copy of our original, the embedded 32-bit

resulting code should match that of our records, and if it is not a copy, we should get general randomness. The third and the fourth possibilities of 3) Is a copy but doesn't match identification number and 4) isn't a copy but does match are, in the case of 3), possible if the signal to noise ratio of the process has plummeted, i.e. the 'suspect image' is truly a very poor copy of the original, and in the case of 4) is basically one chance in four billion since we were using a 32-bit identification number. If we are truly worried about 4), we can just have a second independent lab perform their own tests on a different issue of the same magazine. Finally, checking the error-check bits against what the values give is one final and possibly overkill check on the whole process. In situations where signal to noise is a possible problem, these error checking bits might be eliminated without too much harm.

Benefits

Now that a full description of the first embodiment has been described via a detailed example, it is appropriate to point out the rationale of some of the process steps and their benefits.

The ultimate benefits of the foregoing process are that obtaining an identification number is fully independent of the manners and methods of preparing the difference image. That is to say, the manners of preparing the difference image, such as cutting, registering, scaling, etcetera, cannot increase the odds of finding an identification number when none exists; it only helps the signal-to-noise ratio of the identification process when a true identification number is present. Methods of preparing images for identification can be different from each other even, providing the possibility for multiple independent methodologies for making a match.

The ability to obtain a match even on sub-sets of the original signal or image is a key point in today's information-rich world. Cutting and pasting both images and sound clips is becoming more common, allowing such an embodiment to be used in detecting a copy even when original material has been thus corrupted. Finally, the signal to noise ratio of matching should begin to become difficult only when the copy material itself has been significantly altered either by noise or by significant distortion; both of these also will affect that copy's commercial value, so that trying to thwart the system can only be done at the expense of a huge decrease in commercial value.

An early conception of this technology was the case where only a single "snowy image" or random signal was added to an original image, i.e. the case where $N=1$. "Decoding" this signal would involve a subsequent mathematical analysis using (generally statistical) algorithms to make a judgment on the presence or absence of this signal. The reason this approach was abandoned as the preferred embodiment was that there was an inherent gray area in the certainty of detecting the presence or absence of the signal. By moving onward to a multitude of bit planes, i.e. $N > 1$, combined with simple pre-defined algorithms prescribing the manner of choosing between a "0" and a "1", the certainty question moved from the realm of expert statistical analysis into the realm of guessing a random binary event such

as a coin flip. This is seen as a powerful feature relative to the intuitive acceptance of this technology in both the courtroom and the marketplace. The analogy which summarizes the inventor's thoughts on this whole question is as follows: The search for a single identification signal amounts to calling a coin flip only once, and relying on arcane experts to make the call; whereas the $N > 1$ embodiment relies on the broadly intuitive principle of correctly calling a coin flip N times in a row. This situation is greatly exacerbated, i.e. the problems of "interpretation" of the presence of a single signal, when images and sound clips get smaller and smaller in extent.

Another important reason that the $N > 1$ case is preferred over the $N = 1$ embodiment is that in the $N = 1$ case, the manner in which a suspect image is prepared and manipulated has a direct bearing on the likelihood of making a positive identification. Thus, the manner with which an expert makes an identification determination becomes an integral part of that determination. The existence of a multitude of mathematical and statistical approaches to making this determination leave open the possibility that some tests might make positive identifications while others might make negative determinations, inviting further arcane debate about the relative merits of the various identification approaches. The $N > 1$ embodiment avoids this further gray area by presenting a method where no amount of pre-processing of a signal - other than pre-processing which surreptitiously uses knowledge of the private code signals - can increase the likelihood of "calling the coin flip N times in a row."

The fullest expression of the present system will come when it becomes an industry standard and numerous independent groups set up with their own means or 'in-house' brand of applying embedded identification numbers and in their decipherment. Numerous independent group identification will further enhance the ultimate objectivity of the method, thereby enhancing its appeal as an industry standard.

Use of True Polarity in Creating the Composite Embedded Code Signal

The foregoing discussion made use of the 0 and 1 formalism of binary technology to accomplish its ends. Specifically, the 0's and 1's of the N-bit identification word directly multiplied their corresponding individual embedded code signal to form the composite embedded code signal (step 8, Fig. 2). This approach certainly has its conceptual simplicity, but the multiplication of an embedded code signal by 0 along with the storage of that embedded code contains a kind of inefficiency.

It is preferred to maintain the formalism of the 0 and 1 nature of the N-bit identification word, but to have the 0's of the word induce a subtraction of their corresponding embedded code signal. Thus, in step 8 of Fig. 2, rather than only 'adding' the individual embedded code signals which correspond to a '1' in the N-bit identification word, we will also 'subtract' the individual embedded code signals which correspond to a '0' in the N-bit identification word.

At first glance this seems to add more apparent noise to the final composite signal. But it also increases the energy-wise separation of the 0's from the 1's, and thus the 'gain' which is applied in step 10, Fig. 2 can be correspondingly lower.

We can refer to this improvement as the use of true polarity. The main advantage of this improvement can largely be summarized as 'informational efficiency.'

'Perceptual Orthogonality' of the Individual Embedded Code Signals

The foregoing discussion contemplates the use of generally random noise-like signals as the individual embedded code signals. This is perhaps the simplest form of signal to generate. However, there is a form of informational optimization which can be applied to the set of the individual embedded signals, which the applicant describes under the rubric 'perceptual orthogonality.' This term is loosely based on the mathematical concept of the orthogonality of vectors, with the current additional requirement that this orthogonality should maximize the signal energy of the identification information while maintaining it below some perceptibility threshold. Put another way, the embedded code signals need not necessarily be random in nature.

Use and Improvements of the First Embodiment in the Field of Emulsion-Based Photography

The foregoing discussion outlined techniques that are applicable to photographic materials. The following section explores the details of this area further and discloses certain improvements which lend themselves to a broad range of applications.

The first area to be discussed involves the pre-application or pre-exposing of a serial number onto traditional photographic products, such as negative film, print paper, transparencies, etc. In general, this is a way to embed *a priori* unique serial numbers (and by implication, ownership and tracking information) into photographic material. The serial numbers themselves would be a permanent part of the normally exposed picture, as opposed to being relegated to the margins or stamped on the back of a printed photograph, which all require separate locations and separate methods of copying. The 'serial number' as it is called here is generally synonymous with the N-bit identification word, only now we are using a more common industrial terminology.

In Fig. 2, step 11, the disclosure calls for the storage of the "original [image]" along with code images. Then in Fig. 3, step 9, it directs that the original be subtracted from the suspect image, thereby leaving the possible identification codes plus whatever noise and corruption has accumulated. Therefore, the previous disclosure made the tacit assumption that there exists an original without the composite embedded signals.

Now in the case of selling print paper and other duplication film products, this will still be the case, i.e., an "original" without the embedded codes will indeed exist and the basic methodology of the first embodiment can be employed. The original film serves perfectly well as an 'unencoded original.'

However, in the case where pre-exposed negative film is used, the composite embedded signal pre-exists on the original film and thus there will never be an "original" separate from the pre-embedded signal. It is this latter case, therefore, which will be examined a bit more closely, along with observations on how to best use the principles discussed above (the former cases adhering to the previously outlined methods).

The clearest point of departure for the case of pre-numbered negative film, i.e. negative film which has had each and every frame pre-exposed with a very faint and unique composite embedded signal, comes at step 9 of Fig. 3 as previously noted. There are certainly other differences as well, but they are mostly logistical in nature, such as how and when to embed the signals on the film, how to store the code numbers and serial number, etc. Obviously the pre-exposing of film would involve a major change to the general mass production process of creating and packaging film.

Fig. 4 has a schematic outlining one potential post-hoc mechanism for pre-exposing film. 'Post-hoc' refers to applying a process after the full common manufacturing process of film has already taken place. Eventually, economies of scale may dictate placing this pre-exposing process directly into the chain of manufacturing film. Depicted in Fig. 4 is what is commonly known as a film writing system. The computer, 106, displays the composite signal produced in step 8, Fig. 2, on its phosphor screen. A given frame of film is then exposed by imaging this phosphor screen, where the exposure level is generally very faint, i.e. generally imperceptible. Clearly, the marketplace will set its own demands on how faint this should be, that is, the level of added 'graininess' as practitioners would put it. Each frame of film is sequentially exposed, where in general the composite image displayed on the CRT 102 is changed for each and every frame, thereby giving each frame of film a different serial number. The transfer lens 104 highlights the focal conjugate planes of a film frame and the CRT face.

Getting back to the applying the principles of the foregoing embodiment in the case of pre-exposed negative film... At step 9, Fig. 3, if we were to subtract the "original" with its embedded code, we would obviously be "erasing" the code as well since the code is an integral part of the original. Fortunately, remedies do exist and identifications can still be made. However, it will be a challenge to artisans who refine this embodiment to have the signal to noise ratio of the identification process in the pre-exposed negative case approach the signal to noise ratio of the case where the un-encoded original exists.

A succinct definition of the problem is in order at this point. Given a suspect picture (signal), find the embedded identification code IF a code exists at al. The problem reduces to one of finding the amplitude of each and every individual embedded code signal within the suspect picture, not only within the context of noise and corruption as was previously explained, but now also within the context of the

coupling between a captured image and the codes. 'Coupling' here refers to the idea that the captured image "randomly biases" the cross-correlation.

So, bearing in mind this additional item of signal coupling, the identification process now estimates the signal amplitude of each and every individual embedded code signal (as opposed to taking the cross-correlation result of step 12, Fig. 3). If our identification signal exists in the suspect picture, the amplitudes thus found will split into a polarity with positive amplitudes being assigned a '1' and negative amplitudes being assigned a '0'. Our unique identification code manifests itself. If, on the other hand, no such identification code exists or it is someone else's code, then a random gaussian-like distribution of amplitudes is found with a random hash of values.

It remains to provide a few more details on how the amplitudes of the individual embedded codes are found. Again, fortunately, this exact problem has been treated in other technological applications. Besides, throw this problem and a little food into a crowded room of mathematicians and statisticians and surely a half dozen optimized methodologies will pop out after some reasonable period of time. It is a rather cleanly defined problem.

One specific example solution comes from the field of astronomical imaging. Here, it is a mature prior art to subtract out a "thermal noise frame" from a given CCD image of an object. Often, however, it is not precisely known what scaling factor to use in subtracting the thermal frame, and a search for the correct scaling factor is performed. This is precisely the task of this step of the present embodiment.

General practice merely performs a common search algorithm on the scaling factor, where a scaling factor is chosen and a new image is created according to:

$$\text{NEW IMAGE} = \text{ACQUIRED IMAGE} - \text{SCALE} * \text{THERMAL IMAGE} \qquad (4)$$

The new image is applied to the fast fourier transform routine and a scale factor is eventually found which minimizes the integrated high frequency content of the new image. This general type of search operation with its minimization of a particular quantity is exceedingly common. The scale factor thus found is the sought-for "amplitude." Refinements which are contemplated but not yet implemented are where the coupling of the higher derivatives of the acquired image and the embedded codes are estimated and removed from the calculated scale factor. In other words, certain bias effects from the coupling mentioned earlier are present and should be eventually accounted for and removed both through theoretical and empirical experimentation.

Use and Improvements in the Detection of Signal or Image Alteration

Apart from the basic need of identifying a signal or image as a whole, there is also a rather ubiquitous need to detect possible alterations to a signal or image. The following section describes how the foregoing embodiment, with certain modifications and improvements, can be used as a powerful tool in this area. The potential scenarios and applications of detecting alterations are innumerable.

To first summarize, assume that we have a given signal or image which has been positively identified using the basic methods outlined above. In other words, we know its N-bit identification word, its individual embedded code signals, and its composite embedded code. We can then fairly simply create a spatial map of the composite code's amplitude within our given signal or image. Furthermore, we can divide this amplitude map by the known composite code's spatial amplitude, giving a normalized map, i.e. a map which should fluctuate about some global mean value. By simple examination of this map, we can visually detect any areas which have been significantly altered wherein the value of the normalized amplitude dips below some statistically set threshold based purely on typical noise and corruption (error).

The details of implementing the creation of the amplitude map have a variety of choices. One is to perform the same procedure which is used to determine the signal amplitude as described above, only now we step and repeat the multiplication of any given area of the signal/image with a gaussian weight function centered about the area we are investigating.

Universal Versus Custom Codes

The disclosure thus far has outlined how each and every source signal has its own unique set of individual embedded code signals. This entails the storage of a significant amount of additional code information above and beyond the original, and many applications may merit some form of economizing.

One such approach to economizing is to have a given set of individual embedded code signals be common to a batch of source materials. For example, one thousand images can all utilize the same basic set of individual embedded code signals. The storage requirements of these codes then become a small fraction of the overall storage requirements of the source material.

Furthermore, some applications can utilize a universal set of individual embedded code signals, i.e., codes which remain the same for all instances of distributed material. This type of requirement would be seen by systems which wish to hide the N-bit identification word itself, yet have standardized equipment be able to read that word. This can be used in systems which make go/no go decisions at point-of-read locations. The potential drawback to this set-up is that the universal codes are more prone to be sleuthed or stolen; therefore they will not be as secure as the apparatus and methodology of the previously disclosed arrangement. Perhaps this is just the difference between 'high security' and 'air-tight security,' a distinction carrying little weight with the bulk of potential applications.

Use in Printing, Paper, Documents, Plastic Coated Identification Cards, and Other Material Where Global Embedded Codes Can Be Imprinted

The term 'signal' is often used narrowly to refer to digital data information, audio signals, images, etc. A broader interpretation of 'signal,' and the one more generally intended, includes any form of modulation of any material whatsoever. Thus, the micro-topology of a piece of common paper becomes

a 'signal' (e.g. it height as a function of x-y coordinates). The reflective properties of a flat piece of plastic (as a function of space also) becomes a signal. The point is that photographic emulsions, audio signals, and digitized information are not the only types of signals capable of utilizing the principles described herein.

As a case in point, a machine very much resembling a braille printing machine can be designed so as to imprint unique 'noise-like' indentations as outlined above. These indentations can be applied with a pressure which is much smaller than is typically applied in creating braille, to the point where the patterns are not noticed by a normal user of the paper. But by following the steps of the present disclosure and applying them via the mechanism of micro-indentations, a unique identification code can be placed onto any given sheet of paper, be it intended for everyday stationary purposes, or be it for important documents, legal tender, or other secured material.

The reading of the identification material in such an embodiment generally proceeds by merely reading the document optically at a variety of angles. This would become an inexpensive method for deducing the micro-topology of the paper surface. Certainly other forms of reading the topology of the paper are possible as well.

In the case of plastic encased material such as identification cards, e.g. driver's licenses, a similar braille-like impressions machine can be utilized to imprint unique identification codes. Subtle layers of photoreactive materials can also be embedded inside the plastic and 'exposed.'

It is clear that wherever a material exists which is capable of being modulated by 'noise-like' signals, that material is an appropriate carrier for unique identification codes and utilization of the principles disclosed herein. All that remains is the matter of economically applying the identification information and maintaining the signal level below an acceptability threshold which each and every application will define for itself.

REAL TIME ENCODER

While the first class of embodiments most commonly employs a standard microprocessor or computer to perform the encodation of an image or signal, it is possible to utilize a custom encodation device which may be faster than a typical Von Neuman-type processor. Such a system can be utilized with all manner of serial data streams.

Music and videotape recordings are examples of serial data streams -- data streams which are often pirated. It would assist enforcement efforts if authorized recordings were encoded with identification data so that pirated knock-offs could be traced to the original from which they were made.

Piracy is but one concern driving the need for applicant's technology. Another is authentication. Often it is important to confirm that a given set of data is really what it is purported to be (often several years after its generation).

To address these and other needs, the system 200 of Fig. 5 can be employed. System 200 can be thought of as an identification coding black box 202. The system 200 receives an input signal (sometimes termed the "master" or "unencoded" signal) and a code word, and produces (generally in real time) an identification-coded output signal. (Usually, the system provides key data for use in later decoding.)

The contents of the "black box" 202 can take various forms. An exemplary black box system is shown in Fig. 6 and includes a look-up table 204, a digital noise source 206, first and second scalers 208, 210, an adder/subtracter 212, a memory 214, and a register 216.

The input signal (which in the illustrated embodiment is an 8 - 20 bit data signal provided at a rate of one million samples per second, but which in other embodiments could be an analog signal if appropriate A/D and D/A conversion is provided) is applied from an input 218 to the address input 220 of the look-up table 204. For each input sample (i.e. look-up table address), the table provides a corresponding 8-bit digital output word. This output word is used as a scaling factor that is applied to one input of the first scaler 208.

The first scaler 208 has a second input, to which is applied an 8-bit digital noise signal from source 206. (In the illustrated embodiment, the noise source 206 comprises an analog noise source 222 and an analog-to-digital converter 224 although, again, other implementations can be used.) The noise source in the illustrated embodiment has a zero mean output value, with a full width half maximum (FWHM) of 50 - 100 digital numbers (e.g. from -75 to +75).

The first scaler 208 multiplies the two 8-bit words at its inputs (scale factor and noise) to produce -- for each sample of the system input signal -- a 16-bit output word. Since the noise signal has a zero mean value, the output of the first scaler likewise has a zero mean value.

The output of the first scaler 208 is applied to the input of the second scaler 210. The second scaler serves a global scaling function, establishing the absolute magnitude of the identification signal that will ultimately be embedded into the input data signal. The scaling factor is set through a scale control device 226 (which may take a number of forms, from a simple rheostat to a graphically implemented control in a graphical user interface), permitting this factor to be changed in accordance with the requirements of different applications. The second scaler 210 provides on its output line 228 a scaled noise signal. Each sample of this scaled noise signal is successively stored in the memory 214.

(In the illustrated embodiment, the output from the first scaler 208 may range between -1500 and +1500 (decimal), while the output from the second scaler 210 is in the low single digits, (such as between -2 and +2).)

Register 216 stores a multi-bit identification code word. In the illustrated embodiment this code word consists of 8 bits, although larger code words (up to hundreds of bits) are commonly used. These bits are referenced, one at a time, to control how the input signal is modulated with the scaled noise signal.

In particular, a pointer 230 is cycled sequentially through the bit positions of the code word in register 216 to provide a control bit of "0" or "1" to a control input 232 of the adder/subtracter 212. If, for a particular input signal sample, the control bit is a "1", the scaled noise signal sample on line 232 is added to the input signal sample. If the control bit is a "0", the scaled noise signal sample is subtracted from the input signal sample. The output 234 from the adder/subtracter 212 provides the black box's output signal.

The addition or subtraction of the scaled noise signal in accordance with the bits of the code word effects a modulation of the input signal that is generally imperceptible. However, with knowledge of the contents of the memory 214, a user can later decode the encoding, determining the code number used in the original encoding process. (Actually, use of memory 214 is optional, as explained below.)

It will be recognized that the encoded signal can be distributed in well known ways, including converted to printed image form, stored on magnetic media (floppy diskette, analog or DAT tape, etc.), CD-ROM, etc. etc.

Decoding

A variety of techniques can be used to determine the identification code with which a suspect signal has been encoded. Two are discussed below. The first is less preferable than the latter for most applications, but is discussed herein so that the reader may have a fuller context within which to understand the disclosed technology.

More particularly, the first decoding method is a difference method, relying on subtraction of corresponding samples of the original signal from the suspect signal to obtain difference samples, which are then examined (typically individually) for deterministic coding indicia (i.e. the stored noise data). This approach may thus be termed a "sample-based, deterministic" decoding technique.

The second decoding method does not make use of the original signal. Nor does it examine particular samples looking for predetermined noise characteristics. Rather, the statistics of the suspect signal (or a portion thereof) are considered in the aggregate and analyzed to discern the presence of identification coding that permeates the entire signal. The reference to permeation means the entire identification code can be discerned from a small fragment of the suspect signal. This latter approach may thus be termed a "holographic, statistical" decoding technique.

Both of these methods begin by registering the suspect signal to match the original. This entails scaling (e.g. in amplitude, duration, color balance, etc.), and sampling (or resampling) to restore the original sample rate. As in the earlier described embodiment, there are a variety of well understood techniques by which the operations associated with this registration function can be performed.

As noted, the first decoding approach proceeds by subtracting the original signal from the registered, suspect signal, leaving a difference signal. The polarity of successive difference signal samples

can then be compared with the polarities of the corresponding stored noise signal samples to determine the identification code. That is, if the polarity of the first difference signal sample matches that of the first noise signal sample, then the first bit of the identification code is a "1." (In such case, the polarity of the 9th, 17th, 25th, etc. samples should also all be positive.) If the polarity of the first difference signal sample is opposite that of the corresponding noise signal sample, then the first bit of the identification code is a "0."

By conducting the foregoing analysis with eight successive samples of the difference signal, the sequence of bits that comprise the original code word can be determined. If, as in the illustrated embodiment, pointer 230 stepped through the code word one bit at a time, beginning with the first bit, during encoding, then the first 8 samples of the difference signal can be analyzed to uniquely determine the value of the 8-bit code word.

In a noise-free world (speaking here of noise independent of that with which the identification coding is effected), the foregoing analysis would always yield the correct identification code. But a process that is only applicable in a noise-free world is of limited utility indeed.

(Further, accurate identification of signals in noise-free contexts can be handled in a variety of other, simpler ways: e.g. checksums; statistically improbable correspondence between suspect and original signals; etc.)

While noise-induced aberrations in decoding can be dealt with -- to some degree -- by analyzing large portions of the signal, such aberrations still place a practical ceiling on the confidence of the process. Further, the villain that must be confronted is not always as benign as random noise. Rather, it increasingly takes the form of human-caused corruption, distortion, manipulation, etc. In such cases, the desired degree of identification confidence can only be achieved by other approaches.

The illustrated embodiment (the "holographic, statistical" decoding technique) relies on recombining the suspect signal with certain noise data (typically the data stored in memory 214), and analyzing the entropy of the resulting signal. "Entropy" need not be understood in its most strict mathematical definition, it being merely the most concise word to describe randomness (noise, smoothness, snowiness, etc.).

Most serial data signals are not random. That is, one sample usually correlates -- to some degree -- with the adjacent samples. Noise, in contrast, typically is random. If a random signal (e.g. noise) is added to (or subtracted from) a non-random signal, the entropy of the resulting signal generally increases. That is, the resulting signal has more random variations than the original signal. This is the case with the encoded output signal produced by the present encoding process; it has more entropy than the original, unencoded signal.

If, in contrast, the addition of a random signal to (or subtraction from) a non-random signal reduces entropy, then something unusual is happening. It is this anomaly that the present decoding process uses to detect embedded identification coding.

To fully understand this entropy-based decoding method, it is first helpful to highlight a characteristic of the original encoding process: the similar treatment of every eighth sample.

In the encoding process discussed above, the pointer 230 increments through the code word, one bit for each successive sample of the input signal. If the code word is eight bits in length, then the pointer returns to the same bit position in the code word every eighth signal sample. If this bit is a "1", noise is added to the input signal; if this bit is a "0", noise is subtracted from the input signal. Due to the cyclic progression of the pointer 230, every eighth sample of an encoded signal thus shares a characteristic: they are all either augmented by the corresponding noise data (which may be negative), or they are all diminished, depending on whether the bit of the code word then being addressed by pointer 230 is a "1" or a "0".

To exploit this characteristic, the entropy-based decoding process treats every eighth sample of the suspect signal in like fashion. In particular, the process begins by adding to the 1st, 9th, 17th, 25th, etc. samples of the suspect signal the corresponding scaled noise signal values stored in the memory 214 (i.e. those stored in the 1st, 9th, 17th, 25th, etc., memory locations, respectively). The entropy of the resulting signal (i.e. the suspect signal with every 8th sample modified) is then computed.

(Computation of a signal's entropy or randomness is well understood by artisans in this field. One generally accepted technique is to take the derivative of the signal at each sample point, square these values, and then sum over the entire signal. However, a variety of other well known techniques can alternatively be used.)

The foregoing step is then repeated, this time subtracting the stored noise values from the 1st, 9th, 17th, 25 etc. suspect signal samples.

One of these two operations will undo the encoding process and reduce the resulting signal's entropy; the other will aggravate it. If adding the noise data in memory 214 to the suspect signal reduces its entropy, then this data must earlier have been subtracted from the original signal. This indicates that pointer 230 was pointing to a "0" bit when these samples were encoded. (A "0" at the control input of adder/subtracter 212 caused it to subtract the scaled noise from the input signal.)

Conversely, if subtracting the noise data from every eighth sample of the suspect signal reduces its entropy, then the encoding process must have earlier added this noise. This indicates that pointer 230 was pointing to a "1" bit when samples 1, 9, 17, 25, etc., were encoded.

By noting whether entropy decreases by (a) adding or (b) subtracting the stored noise data to/from the suspect signal, it can be determined that the first bit of the code word is (a) a "0", or (b) a "1".

The foregoing operations are then conducted for the group of spaced samples of the suspect signal beginning with the second sample (i.e. 2, 10, 18, 26 ...). The entropy of the resulting signals indicate whether the second bit of the code word is a "0" or a "1". Likewise with the following 6 groups of spaced samples in the suspect signal, until all 8 bits of the code word have been discerned.

It will be appreciated that the foregoing approach is not sensitive to corruption mechanisms that alter the values of individual samples; instead, the process considers the entropy of the signal as a whole, yielding a high degree of confidence in the results. Further, even small excerpts of the signal can be analyzed in this manner, permitting piracy of even small details of an original work to be detected. The results are thus statistically robust, both in the face of natural and human corruption of the suspect signal.

It will further be appreciated that the use of an N-bit code word in this real time embodiment provides benefits analogous to those discussed above in connection with the batch encoding system. (Indeed, the present embodiment may be conceptualized as making use of N different noise signals, just as in the batch encoding system. The first noise signal is a signal having the same extent as the input signal, and comprising the scaled noise signal at the 1st, 9th, 17th, 25th, etc., samples (assuming N=8), with zeroes at the intervening samples. The second noise signal is a similar one comprising the scaled noise signal at the 2d, 10th, 18th, 26th, etc., samples, with zeroes at the intervening samples. Etc. These signals are all combined to provide a composite noise signal.) One of the important advantages inherent in such a system is the high degree of statistical confidence (confidence which doubles with each successive bit of the identification code) that a match is really a match. The system does not rely on subjective evaluation of a suspect signal for a single, deterministic embedded code signal.

Illustrative Variations

From the foregoing description, it will be recognized that numerous modifications can be made to the illustrated systems without changing the fundamental principles. A few of these variations are described below.

The above-described decoding process tries both adding and subtracting stored noise data to/from the suspect signal in order to find which operation reduces entropy. In other embodiments, only one of these operations needs to be conducted. For example, in one alternative decoding process the stored noise data corresponding to every eighth sample of the suspect signal is only added to said samples. If the entropy of the resulting signal is thereby increased, then the corresponding bit of the code word is a "1" (i.e. this noise was added earlier, during the encoding process, so adding it again only compounds the signal's randomness). If the entropy of the resulting signal is thereby decreased, then the corresponding bit of the code word is a "0". A further test of entropy if the stored noise samples are subtracted is not required.

The statistical reliability of the identification process (coding and decoding) can be designed to exceed virtually any confidence threshold (e.g. 99.9%, 99.99%, 99.999%, etc. confidence) by appropriate selection of the global scaling factors, etc. Additional confidence in any given application (unnecessary in most applications) can be achieved by rechecking the decoding process.

One way to recheck the decoding process is to remove the stored noise data from the suspect signal in accordance with the bits of the discerned code word, yielding a "restored" signal (e.g. if the first bit of the code word is found to be "1," then the noise samples stored in the 1st, 9th, 17th, etc. locations of the memory 214 are subtracted from the corresponding samples of the suspect signal). The entropy of the restored signal is measured and used as a baseline in further measurements. Next, the process is repeated, this time removing the stored noise data from the suspect signal in accordance with a modified code word. The modified code word is the same as the discerned code word, except 1 bit is toggled (e.g. the first). The entropy of the resulting signal is determined, and compared with the baseline. If the toggling of the bit in the discerned code word resulted in increased entropy, then the accuracy of that bit of the discerned code word is confirmed. The process repeats, each time with a different bit of the discerned code word toggled, until all bits of the code word have been so checked. Each change should result in an increase in entropy compared to the baseline value.

The data stored in memory 214 is subject to a variety of alternatives. In the foregoing discussion, memory 214 contains the scaled noise data. In other embodiments, the unscaled noise data can be stored instead.

In still other embodiments, it can be desirable to store at least part of the input signal itself in memory 214. For example, the memory can allocate 8 signed bits to the noise sample, and 16 bits to store the most significant bits of an 18- or 20-bit audio signal sample. This has several benefits. One is that it simplifies registration of a "suspect" signal. Another is that, in the case of encoding an input signal which was already encoded, the data in memory 214 can be used to discern which of the encoding processes was performed first. That is, from the input signal data in memory 214 (albeit incomplete), it is generally possible to determine with which of two code words it has been encoded.

Yet another alternative for memory 214 is that is can be omitted altogether.

One way this can be achieved is to use a deterministic noise source in the encoding process, such as an algorithmic noise generator seeded with a known key number. The same deterministic noise source, seeded with the same key number, can be used in the decoding process. In such an arrangement, only the key number needs be stored for later use in decoding, instead of the large data set usually stored in memory 214.

Alternatively, if the noise signal added during encoding does not have a zero mean value, and the length N of the code word is known to the decoder, then a universal decoding process can be implemented. This process uses the same entropy test as the foregoing procedures, but cycles through possible code

words, adding/subtracting a small dummy noise value (e.g. less than the expected mean noise value) to every Nth sample of the suspect signal, in accordance with the bits of the code word being tested, until a reduction in entropy is noted. Such an approach is not favored for most applications, however, because it offers less security than the other embodiments (e.g. it is subject to cracking by brute force).

Many applications are well served by the embodiment illustrated in Fig. 7, in which different code words are used to produce several differently encoded versions of an input signal, each making use of the same noise data. More particularly, the embodiment 240 of Fig. 7 includes a noise store 242 into which noise from source 206 is written during the identification-coding of the input signal with a first code word. (The noise source of Fig. 7 is shown outside of the real time encoder 202 for convenience of illustration.) Thereafter, additional identification-coded versions of the input signal can be produced by reading the stored noise data from the store and using it in conjunction with second through Nth code words to encode the signal. (While binary-sequential code words are illustrated in Fig. 7, in other embodiments arbitrary sequences of code words can be employed.) With such an arrangement, a great number of differently-encoded signals can be produced, without requiring a proportionally-sized long term noise memory. Instead, a fixed amount of noise data is stored, whether encoding an original once or a thousand times.

(If desired, several differently-coded output signals can be produced at the same time, rather than seriatim. One such implementation includes a plurality of adder/subtracter circuits 212, each driven with the same input signal and with the same scaled noise signal, but with different code words. Each, then, produces a differently encoded output signal.)

In applications having a great number of differently-encoded versions of the same original, it will be recognized that the decoding process need not always discern every bit of the code word. Sometimes, for example, the application may require identifying only a group of codes to which the suspect signal belongs. (E.g., high order bits of the code word might indicate an organization to which several differently coded versions of the same source material were provided, with low-order bits identifying specific copies. To identify the organization with which a suspect signal is associated, it may not be necessary to examine the low order bits, since the organization can be identified by the high order bits alone.) If the identification requirements can be met by discerning a subset of the code word bits in the suspect signal, the decoding process can be shortened.

Some applications may be best served by restarting the encoding process -- sometimes with a different code word -- several times within an integral work. Consider, as an example, videotaped productions (e.g. television programming). Each frame of a videotaped production can be identification-coded with a unique code number, processed in real-time with an arrangement 248 like that shown in Fig. 8. Each time a vertical retrace is detected by sync detector 250, the noise source 206 resets (e.g. to repeat the sequence just produced) and an identification code increments to the next value. Each frame of the

videotape is thereby uniquely identification-coded. Typically, the encoded signal is stored on a videotape for long term storage (although other storage media, including laser disks, can be used).

Returning to the encoding apparatus, the look-up table 204 in the illustrated embodiment exploits the fact that high amplitude samples of the input data signal can tolerate (without objectionable degradation of the output signal) a higher level of encoded identification coding than can low amplitude input samples. Thus, for example, input data samples having decimal values of 0, 1 or 2 may be correspond (in the look-up table 204) to scale factors of unity (or even zero), whereas input data samples having values in excess of 200 may correspond to scale factors of 15. Generally speaking, the scale factors and the input sample values correspond by a square root relation. That is, a four-fold increase in a value of the sampled input signal corresponds to approximately a two-fold increase in a value of the scaling factor associated therewith.

(The parenthetical reference to zero as a scaling factor alludes to cases, e.g., in which the source signal is temporally or spatially devoid of information content. In an image, for example, a region characterized by several contiguous sample values of zero may correspond to a jet black region of the frame. A scaling value of zero may be appropriate here since there is essentially no image data to be pirated.)

Continuing with the encoding process, those skilled in the art will recognized the potential for "rail errors" in the illustrated embodiment. For example, if the input signal consists of 8-bit samples, and the samples span the entire range from 0 to 255 (decimal), then the addition or subtraction of scaled noise to/from the input signal may produce output signals that cannot be represented by 8 bits (e.g. -2, or 257). A number of well-understood techniques exist to rectify this situation, some of them proactive and some of them reactive. (Among these known techniques are: specifying that the input signal shall not have samples in the range of 0-4 or 251-255, thereby safely permitting modulation by the noise signal; or including provision for detecting and adaptively modifying input signal samples that would otherwise cause rail errors.)

While the illustrated embodiment describes stepping through the code word sequentially, one bit at a time, to control modulation of successive bits of the input signal, it will be appreciated that the bits of the code word can be used other than sequentially for this purpose. Indeed, bits of the code word can be selected in accordance with any predetermined algorithm.

The dynamic scaling of the noise signal based on the instantaneous value of the input signal is an optimization that can be omitted in many embodiments. That is, the look-up table 204 and the first scaler 208 can be omitted entirely, and the signal from the digital noise source 206 applied directly (or through the second, global scaler 210) to the adder/subtracter 212.

It will be further recognized that the use of a zero-mean noise source simplifies the illustrated embodiment, but is not essential. A noise signal with another mean value can readily be used, and D.C. compensation (if needed) can be effected elsewhere in the system.

The use of a noise source 206 is also optional. A variety of other signal sources can be used, depending on application- dependent constraints (e.g. the threshold at which the encoded identification signal becomes perceptible). In many instances, the level of the embedded identification signal is low enough that the identification signal needn't have a random aspect; it is imperceptible regardless of its nature. A pseudo random source 206, however, is usually desired because it provides the greatest identification code signal S/N ratio (a somewhat awkward term in this instance) for a level of imperceptibility of the embedded identification signal.

It will be recognized that identification coding need not occur after a signal has been reduced to stored form as data (i.e. "fixed in tangible form," in the words of the U.S. Copyright Act). Consider, for example, the case of popular musicians whose performances are often recorded illicitly. By identification coding the audio before it drives concert hall speakers, unauthorized recordings of the concert can be traced to a particular place and time. Likewise, live audio sources such as 911 emergency calls can be encoded prior to recording so as to facilitate their later authentication.

While the black box embodiment has been described as a stand alone unit, it will be recognized that it can be integrated into a number of different tools/instruments as a component. One is a scanner, which can embed identification codes in the scanned output data. (The codes can simply serve to memorialize that the data was generated by a particular scanner). Another is in creativity software, such as popular drawing/graphics/animation/paint programs offered by Adobe, Macromedia, Corel, and the like.

Finally, while the real-time encoder 202 has been illustrated with reference to a particular hardware implementation, it will be recognized that a variety of other implementations can alternatively be employed. Some utilize other hardware configurations. Others make use of software routines for some or all of the illustrated functional blocks. (The software routines can be executed on any number of different general purpose programmable computers, such as 80x86 PC-compatible computers, RISC-based workstations, etc.)

TYPES OF NOISE, QUASI-NOISE, AND OPTIMIZED-NOISE

Heretofore this disclosure postulated Gaussian noise, "white noise," and noise generated directly from application instrumentation as a few of the many examples of the kind of carrier signal appropriate to carry a single bit of information throughout an image or signal. It is possible to be even more proactive in "designing" characteristics of noise in order to achieve certain goals. The "design" of using Gaussian or instrumental noise was aimed somewhat toward "absolute" security. This section of the disclosure takes a

look at other considerations for the design of the noise signals which may be considered the ultimate carriers of the identification information.

For some applications it might be advantageous to design the noise carrier signal (e.g. the Nth embedded code signal in the first embodiment; the scaled noise data in the second embodiment), so as to provide more absolute signal strength to the identification signal relative to the perceptibility of that signal. One example is the following. It is recognized that a true Gaussian noise signal has the value '0' occur most frequently, followed by 1 and -1 at equal probabilities to each other but lower than '0', 2 and -2 next, and so on. Clearly, the value zero carries no information as it is used in such an embodiment. Thus, one simple adjustment, or design, would be that any time a zero occurs in the generation of the embedded code signal, a new process takes over, whereby the value is converted "randomly" to either a 1 or a -1. In logical terms, a decision would be made: if '0', then random(1,-1). The histogram of such a process would appear as a Gaussian/Poissonian type distribution, except that the 0 bin would be empty and the 1 and -1 bin would be increased by half the usual histogram value of the 0 bin.

In this case, identification signal energy would always be applied at all parts of the signal. A few of the trade-offs include: there is a (probably negligible) lowering of security of the codes in that a "deterministic component" is a part of generating the noise signal. The reason this might be completely negligible is that we still wind up with a coin flip type situation on randomly choosing the 1 or the -1. Another trade-off is that this type of designed noise will have a higher threshold of perceptibility, and will only be applicable to applications where the least significant bit of a data stream or image is already negligible relative to the commercial value of the material, i.e. if the least significant bit were stripped from the signal (for all signal samples), no one would know the difference and the value of the material would not suffer. This blocking of the zero value in the example above is but one of many ways to "optimize" the noise properties of the signal carrier, as anyone in the art can realize. We refer to this also as "quasi-noise" in the sense that natural noise can be transformed in a pre-determined way into signals which for all intents and purposes will read as noise. Also, cryptographic methods and algorithms can easily, and often by definition, create signals which are perceived as completely random. Thus the word "noise" can have different connotations, primarily between that as defined subjectively by an observer or listener, and that defined mathematically. The difference of the latter is that mathematical noise has different properties of security and the simplicity with which it can either be "sleuthed" or the simplicity with which instruments can "automatically recognize" the existence of this noise.

"Universal" Embedded Codes

The bulk of this disclosure teaches that for absolute security, the noise-like embedded code signals which carry the bits of information of the identification signal should be unique to each and every encoded signal, or, slightly less restrictive, that embedded code signals should be generated sparingly, such as using the

same embedded codes for a batch of 1000 pieces of film, for example. Be this as it may, there is a whole other approach to this issue wherein the use of what we will call "universal" embedded code signals can open up large new applications for this technology. The economics of these uses would be such that the de facto lowered security of these universal codes (e.g. they would be analyzable by time honored cryptographic decoding methods, and thus potentially thwarted or reversed) would be economically negligible relative to the economic gains that the intended uses would provide. Piracy and illegitimate uses would become merely a predictable "cost" and a source of uncollected revenue only; a simple line item in an economic analysis of the whole. A good analogy of this is in the cable industry and the scrambling of video signals. Everybody seems to know that crafty, skilled technical individuals, who may be generally law abiding citizens, can climb a ladder and flip a few wires in their cable junction box in order to get all the pay channels for free. The cable industry knows this and takes active measures to stop it and prosecute those caught, but the "lost revenue" derived from this practice remains prevalent but almost negligible as a percentage of profits gained from the scrambling system as a whole. The scrambling system as a whole is an economic success despite its lack of "absolute security."

The same holds true for applications of this technology wherein, for the price of lowering security by some amount, large economic opportunity presents itself. This section first describes what is meant by universal codes, then moves on to some of the interesting uses to which these codes can be applied.

Universal embedded codes generally refer to the idea that knowledge of the exact codes can be distributed. The embedded codes won't be put into a dark safe never to be touched until litigation arises (as alluded to in other parts of this disclosure), but instead will be distributed to various locations where on-the-spot analysis can take place. Generally this distribution will still take place within a security controlled environment, meaning that steps will be taken to limit the knowledge of the codes to those with a need to know. Instrumentation which attempts to automatically detect copyrighted material is a non-human example of "something" with a need to know the codes.

There are many ways to implement the idea of universal codes, each with their own merits regarding any given application. For the purposes of teaching this art, we separate these approaches into three broad categories: universal codes based on libraries, universal codes based on deterministic formula, and universal codes based on pre-defined industry standard patterns. A rough rule of thumb is that the first is more secure than the latter two, but that the latter two are possibly more economical to implement than the first.

Universal Codes: 1) Libraries of Universal Codes

The use of libraries of universal codes simply means that applicant's techniques are employed as described, except for the fact that only a limited set of the individual embedded code signals are generated and that any given encoded material will make use of some sub-set of this limited "universal set." An

example is in order here. A photographic print paper manufacturer may wish to pre-expose every piece of 8 by 10 inch print paper which they sell with a unique identification code. They also wish to sell identification code recognition software to their large customers, service bureaus, stock agencies, and individual photographers, so that all these people can not only verify that their own material is correctly marked, but so that they can also determine if third party material which they are about to acquire has been identified by this technology as being copyrighted. This latter information will help them verify copyright holders and avoid litigation, among many other benefits. In order to "economically" institute this plan, they realize that generating unique individual embedded codes for each and every piece of print paper would generate Terabytes of independent information, which would need storing and to which recognition software would need access. Instead, they decide to embed their print paper with 16 bit identification codes derived from a set of only 50 independent "universal" embedded code signals. The details of how this is done are in the next paragraph, but the point is that now their recognition software only needs to contain a limited set of embedded codes in their library of codes, typically on the order of 1 Megabyte to 10 Megabytes of information for 50x16 individual embedded codes splayed out onto an 8x10 photographic print (allowing for digital compression). The reason for picking 50 instead of just 16 is one of a little more added security, where if it were the same 16 embedded codes for all photographic sheets, not only would the serial number capability be limited to 2 to the 16th power, but lesser and lesser sophisticated pirates could crack the codes and remove them using software tools.

There are many different ways to implement this scheme, where the following is but one exemplary method. It is determined by the wisdom of company management that a 300 pixels per inch criteria for the embedded code signals is sufficient resolution for most applications. This means that a composite embedded code image will contain 3000 pixels by 2400 pixels to be exposed at a very low level onto each 8x10 sheet. This gives 7.2 million pixels. Using our staggered coding system such as described in the black box implementation of Figs. 5 and 6, each individual embedded code signal will contain only 7.2 million divided by 16, or approximately 450K true information carrying pixels, i.e. every 16th pixel along a given raster line. These values will typically be in the range of 2 to -2 in digital numbers, or adequately described by a signed 3 bit number. The raw information content of an embedded code is then approximately 3/8th's bytes times 450K or about 170 Kilobytes. Digital compression can reduce this further. All of these decisions are subject to standard engineering optimization principles as defined by any given application at hand, as is well known in the art. Thus we find that 50 of these independent embedded codes will amount to a few Megabytes. This is quite reasonable level to distribute as a "library" of universal codes within the recognition software. Advanced standard encryption devices could be employed to mask the exact nature of these codes if one were concerned that would-be pirates would buy the recognition software merely to reverse engineer the universal embedded codes. The recognition

software could simply unencrypt the codes prior to applying the recognition techniques taught in this disclosure.

The recognition software itself would certainly have a variety of features, but the core task it would perform is determining if there is some universal copyright code within a given image. The key questions become WHICH 16 of the total 50 universal codes it might contain, if any, and if there are 16 found, what are their bit values. The key variables in determining the answers to these questions are: registration, rotation, magnification (scale), and extent. In the most general case with no helpful hints whatsoever, all variables must be independently varied across all mutual combinations, and each of the 50 universal codes must then be checked by adding and subtracting to see if an entropy decrease occurs. Strictly speaking, this is an enormous job, but many helpful hints will be found which make the job much simpler, such as having an original image to compare to the suspected copy, or knowing the general orientation and extent of the image relative to an 8x10 print paper, which then through simple registration techniques can determine all of the variables to some acceptable degree. Then it merely requires cycling through the 50 universal codes to find any decrease in entropy. If one does, then 15 others should as well. A protocol needs to be set up whereby a given order of the 50 translates into a sequence of most significant bit through least significant bit of the ID code word. Thus if we find that universal code number "4" is present, and we find its bit value to be "0", and that universal codes "1" through "3" are definitely not present, then our most significant bit of our N-bit ID code number is a "0". Likewise, we find that the next lowest universal code present is number "7" and it turns out to be a "1", then our next most significant bit is a "1". Done properly, this system can cleanly trace back to the copyright owner so long as they registered their photographic paper stock serial number with some registry or with the manufacturer of the paper itself. That is, we look up in the registry that a paper using universal embedded codes 4,7,11,12,15,19,21,26,27,28,34,35,37,38,40, and 48, and having the embedded code 0110 0101 0111 0100 belongs to Leonardo de Boticelli, an unknown wildlife photographer and glacier cinematographer whose address is in Northern Canada. We know this because he dutifully registered his film and paper stock, a few minutes of work when he bought the stock, which he plopped into the "no postage necessary" envelope that the manufacturing company kindly provided to make the process ridiculously simple. Somebody owes Leonardo a royalty check it would appear, and certainly the registry has automated this royalty payment process as part of its services.

One final point is that truly sophisticated pirates and others with illicit intentions can indeed employ a variety of cryptographic and not so cryptographic methods to crack these universal codes, sell them, and make software and hardware tools which can assist in the removing or distorting of codes. We shall not teach these methods as part of this disclosure, however. In any event, this is one of the prices which must be paid for the ease of universal codes and the applications they open up.

Universal Codes: 2) Universal Codes Based on Deterministic Formulas

The libraries of universal codes require the storage and transmittal of Megabytes of independent, generally random data as the keys with which to unlock the existence and identity of signals and imagery that have been marked with universal codes. Alternatively, various deterministic formulas can be used which "generate" what appear to be random data/image frames, thereby obviating the need to store all of these codes in memory and interrogate each and of the "50" universal codes. Deterministic formulas can also assist in speeding up the process of determining the ID code once one is known to exist in a given signal or image. On the other hand, deterministic formulas lend themselves to sleuthing by less sophisticated pirates. And once sleuthed, they lend themselves to easier communication, such as posting on the Internet to a hundred newsgroups. There may well be many applications which do not care about sleuthing and publishing, and deterministic formulas for generating the individual universal embedded codes might be just the ticket.

Universal Codes: 3) "Simple" Universal Codes

This category is a bit of a hybrid of the first two, and is most directed at truly large scale implementations of the principles of this technology. The applications employing this class are of the type where staunch security is much less important than low cost, large scale implementation and the vastly larger economic benefits that this enables. One exemplary application is placement of identification recognition units directly within modestly priced home audio and video instrumentation (such as a TV). Such recognition units would typically monitor audio and/or video looking for these copyright identification codes, and thence triggering simple decisions based on the findings, such as disabling or enabling recording capabilities, or incrementing program specific billing meters which are transmitted back to a central audio/video service provider and placed onto monthly invoices. Likewise, it can be foreseen that "black boxes" in bars and other public places can monitor (listen with a microphone) for copyrighted materials and generate detailed reports, for use by ASCAP, BMI, and the like.

A core principle of simple universal codes is that some basic industry standard "noiselike" and seamlessly repetitive patterns are injected into signals, images, and image sequences so that inexpensive recognition units can either A) determine the mere existence of a copyright "flag", and B) additionally to A, determine precise identification information which can facilitate more complex decision making and actions.

In order to implement this particular embodiment, the basic principles of generating the individual embedded noise signals need to be simplified in order to accommodate inexpensive recognition signal processing circuitry, while maintaining the properties of effective randomness and holographic permeation. With large scale industry adoption of these simple codes, the codes themselves would border on public domain information (much as cable scrambling boxes are almost de facto public domain), leaving the door

open for determined pirates to develop black market countermeasures, but this situation would be quite analogous to the scrambling of cable video and the objective economic analysis of such illegal activity.

One prior art known to the applicant in this general area of pro-active copyright detection is the Serial Copy Management System adopted by many firms in the audio industry. To the best of applicant's knowledge, this system employs a non-audio "flag" signal which is not part of the audio data stream, but which is nevertheless grafted onto the audio stream and can indicate whether the associated audio data should or should not be duplicated. One problem with this system is that it is restricted to media and instrumentation which can support this extra "flag" signal. Another deficiency is that the flagging system carries no identity information which would be useful in making more complex decisions. Yet another difficulty is that high quality audio sampling of an analog signal can come arbitrarily close to making a perfect digital copy of some digital master and there seems to be no provision for inhibiting this possibility.

Applicant's technology can be brought to bear on these and other problems, in audio applications, video, and all of the other applications previously discussed. An exemplary application of simple universal codes is the following. A single industry standard "1.000000 second of noise" would be defined as the most basic indicator of the presence or absence of the copyright marking of any given audio signal. Fig. 9 has an example of what the waveform of an industry standard noise second might look like, both in the time domain 400 and the frequency domain 402. It is by definition a continuous function and would adapt to any combination of sampling rates and bit quantizations. It has a normalized amplitude and can be scaled arbitrarily to any digital signal amplitude. The signal level and the first M'th derivatives of the signal are continuous at the two boundaries 404 (Fig. 9C), such that when it is repeated, the "break" in the signal would not be visible (as a waveform) or audible when played through a high end audio system. The choice of 1 second is arbitrary in this example, where the precise length of the interval will be derived from considerations such as audibility, quasi-white noise status, seamless repeatability, simplicity of recognition processing, and speed with which a copyright marking determination can be made. The injection of this repeated noise signal onto a signal or image (again, at levels below human perception) would indicate the presence of copyright material. This is essentially a one bit identification code, and the embedding of further identification information will be discussed later on in this section. The use of this identification technique can extend far beyond the low cost home implementations discussed here, where studios could use the technique, and monitoring stations could be set up which literally monitor hundreds of channels of information simultaneously, searching for marked data streams, and furthermore searching for the associated identity codes which could be tied in with billing networks and royalty tracking systems.

This basic, standardized noise signature is seamlessly repeated over and over again and added to audio signals which are to be marked with the base copyright identification. Part of the reason for the word "simple" is seen here: clearly pirates will know about this industry standard signal, but their illicit

uses derived from this knowledge, such as erasure or corruption, will be economically minuscule relative to the economic value of the overall technique to the mass market. For most high end audio this signal will be some 80 to 100 dB down from full scale, or even much further; each situation can choose its own levels though certainly there will be recommendations. The amplitude of the signal can be modulated according to the audio signal levels to which the noise signature is being applied, i.e. the amplitude can increase significantly when a drum beats, but not so dramatically as to become audible or objectionable. These measures merely assist the recognition circuitry to be described.

Recognition of the presence of this noise signature by low cost instrumentation can be effected in a variety of ways. One rests on basic modifications to the simple principles of audio signal power metering. Software recognition programs can also be written, and more sophisticated mathematical detection algorithms can be applied to audio in order to make higher confidence detection identifications. In such embodiments, detection of the copyright noise signature involves comparing the time averaged power level of an audio signal with the time averaged power level of that same audio signal which has had the noise signature subtracted from it. If the audio signal with the noise signature subtracted has a lower power level that the unchanged audio signal, then the copyright signature is present and some status flag to that effect needs to be set. The main engineering subtleties involved in making this comparison include: dealing with audio speed playback discrepancies (e.g. an instrument might be 0.5% "slow" relative to exactly one second intervals); and, dealing with the unknown phase of the one second noise signature within any given audio (basically, its "phase" can be anywhere from 0 to 1 seconds). Another subtlety, not so central as the above two but which nonetheless should be addressed, is that the recognition circuits should not subtract a higher amplitude of the noise signature than was originally embedded onto the audio signal. Fortunately this can be accomplished by merely subtracting only a small amplitude of the noise signal, and if the power level goes down, this is an indication of "heading toward a trough" in the power levels. Yet another related subtlety is that the power level changes will be very small relative to the overall power levels, and calculations generally will need to be done with appropriate bit precision, e.g. 32 bit value operations and accumulations on 16-20 bit audio in the calculations of time averaged power levels.

Clearly, designing and packaging this power level comparison processing circuitry for low cost applications is an engineering optimization task. One trade-off will be the accuracy of making an identification relative to the "short-cuts" which can be made to the circuitry in order to lower its cost and complexity. One embodiment for placing this recognition circuitry inside of instrumentation is through a single programmable integrated circuit which is custom made for the task. Fig. 10 shows one such integrated circuit 506. Here the audio signal comes in, 500, either as a digital signal or as an analog signal to be digitized inside the IC 500, and the output is a flag 502 which is set to one level if the copyright noise signature is found, and to another level if it is not found. Also depicted is the fact that the

standardized noise signature waveform is stored in Read Only Memory, 504, inside the IC 506. There will be a slight time delay between the application of an audio signal to the IC 506 and the output of a valid flag 502, due to the need to monitor some finite portion of the audio before a recognition can place. In this case, there may need to be a "flag valid" output 508 where the IC informs the external world if it has had enough time to make a proper determination of the presence or absence of the copyright noise signature.

There are a wide variety of specific designs and philosophies of designs applied to accomplishing the basic function of the IC 506 of Fig. 10. Audio engineers and digital signal processing engineers are able to generate several fundamentally different designs. One such design is depicted in Fig. 11 by a process 599, which itself is subject to further engineering optimization as will be discussed. Fig. 11 depicts a flow chart for any of: an analog signal processing network, a digital signal processing network, or programming steps in a software program. We find an input signal 600 which along one path is applied to a time averaged power meter 602, and the resulting power output itself treated as a signal P_{in}. To the upper right we find the standard noise signature 504 which will be read out at 125% of normal speed, 604, thus changing its pitch, giving the "pitch changed noise signal" 606. Then the input signal has this pitch changed noise signal subtracted in step 608, and this new signal is applied to the same form of time averaged power meter as in 602, here labelled 610. The output of this operation is also a time based signal here labelled as $P_{s\text{-}pcn}$, 610. Step 612 then subtracts the power signal 602 from the power signal 610, giving an output difference signal P_{out}, 613. If the universal standard noise signature does indeed exist on the input audio signal 600, then case 2, 616, will be created wherein a beat signal 618 of approximately 4 second period will show up on the output signal 613, and it remains to detect this beat signal with a step such as in Fig. 12, 622. Case 1, 614, is a steady noisy signal which exhibits no periodic beating. 125% at step 604 is chosen arbitrarily here, where engineering considerations would determine an optimal value, leading to different beat signal frequencies 618. Whereas waiting 4 seconds in this example would be quite a while, especially is you would want to detect at least two or three beats, Fig. 12 outlines how the basic design of Fig. 11 could be repeated and operated upon various delayed versions of the input signal, delayed by something like 1/20th of a second, with 20 parallel circuits working in concert each on a segment of the audio delayed by 0.05 seconds from their neighbors. In this way, a beat signal will show up approximately every 1/5th of a second and will look like a travelling wave down the columns of beat detection circuits. The existence or absence of this travelling beat wave triggers the detection flag 502. Meanwhile, there would be an audio signal monitor 624 which would ensure that, for example, at least two seconds of audio has been heard before setting the flag valid signal 508.

Though the audio example was described above, it should be clear to anyone in the art that the same type of definition of some repetitive universal noise signal or image could be applied to the many other signals, images, pictures, and physical media already discussed.

The above case deals only with a single bit plane of information, i.e., the noise signature signal is either there (1) or it isn't (0). For many applications, it would be nice to detect serial number information as well, which could then be used for more complex decisions, or for logging information on billing statements or whatnot. The same principles as the above would apply, but now there would be N independent noise signatures as depicted in Fig. 9 instead one single such signature. Typically, one such signature would be the master upon which the mere existence of a copyright marking is detected, and this would have generally higher power than the others, and then the other lower power "identification" noise signatures would be embedded into audio. Recognition circuits, once having found the existence of the primary noise signature, would then step through the other N noise signatures applying the same steps as described above. Where a beat signal is detected, this indicates the bit value of '1', and where no beat signal is detected, this indicates a bit value of '0'. It might be typical that N will equal 32, that way 2^{32} number of identification codes are available to any given industry employing this technology.

Use of this Technology When the Length of the Identification Code is 1

The principles described herein can obviously be applied in the case where only a single presence or absence of an identification signal -- a fingerprint if you will -- is used to provide confidence that some signal or image is copyrighted. The example above of the industry standard noise signature is one case in point. We no longer have the added confidence of the coin flip analogy, we no longer have tracking code capabilities or basic serial number capabilities, but many applications may not require these attributes and the added simplicity of a single fingerprint might outweigh these other attributes in any event.

The "Wallpaper" Analogy

The term "holographic" has been used in this disclosure to describe how an identification code number is distributed in a largely integral form throughout an encoded signal or image. This also refers to the idea that any given fragment of the signal or image contains the entire unique identification code number. As with physical implementations of holography, there are limitations on how small a fragment can become before one begins to lose this property, where the resolution limits of the holographic media are the main factor in this regard for holography itself. In the case of an uncorrupted distribution signal which has used the encoding device of Fig. 5, and which furthermore has used our "designed noise" of above wherein the zero's were randomly changed to a 1 or -1, then the extent of the fragment required is merely N contiguous samples in a signal or image raster line, where N is as defined previously being the length of our identification code number. This is an informational extreme; practical situations where noise and corruption are operative will require generally one, two or higher orders of magnitude more samples than this simple number N. Those skilled in the art will recognize that there are many variables

involved in pinning down precise statistics on the size of the smallest fragment with which an identification can be made.

For tutorial purposes, the applicant also uses the analogy that the unique identification code number is "wallpapered" across and image (or signal). That is, it is repeated over and over again all throughout an image. This repetition of the ID code number can be regular, as in the use of the encoder of Fig. 5, or random itself, where the bits in the ID code 216 of Fig. 6 are not stepped through in a normal repetitive fashion but rather are randomly selected on each sample, and the random selection stored along with the value of the output 228 itself. in any event, the information carrier of the ID code, the individual embedded code signal, does change across the image or signal. Thus as the wallpaper analogy summarizes: the ID code repeats itself over and over, but the patterns that each repetition imprints change randomly accordingly to a generally unsleuthable key.

Lossy Data Compression

As earlier mentioned, applicant's preferred forms of identification coding withstand lossy data compression, and subsequent decompression. Such compression is finding increasing use, particularly in contexts such as the mass distribution of digitized entertainment programming (movies, etc.).

While data encoded according to the disclosed techniques can withstand all types of lossy compression known to applicant, those expected to be most commercially important are the CCITT G3, CCITT G4, JPEG, MPEG and JBIG compression/decompression standards. The CCITT standards are widely used in black-and-white document compression (e.g. facsimile and document-storage). JPEG is most widely used with still images. MPEG is most widely used with moving images. JBIG is a likely successor to the CCITT standards for use with black-and-white imagery. Such techniques are well known to those in the lossy data compression field; a good overview can be found in Pennebaker et al, *JPEG, Still Image Data Compression Standard*, Van Nostrand Reinhold, N.Y., 1993.

Towards Steganography Proper and the Use of this Technology in Passing More Complex Messages or Information

This disclosure concentrates on what above was called wallpapering a single identification code across an entire signal. This appears to be a desirable feature for many applications. However, there are other applications where it might be desirable to pass messages or to embed very long strings of pertinent identification information in signals and images. One of many such possible applications would be where a given signal or image is meant to be manipulated by several different groups, and that certain regions of an image are reserved for each group's identification and insertion of pertinent manipulation information.

In these cases, the code word 216 in Fig. 6 can actually change in some pre-defined manner as a function of signal or image position. For example, in an image, the code could change for each and every

raster line of the digital image. It might be a 16 bit code word, 216, but each scan line would have a new code word, and thus a 480 scan line image could pass a 980 (480 x 2 bytes) byte message. A receiver of the message would need to have access to either the noise signal stored in memory 214, or would have to know the universal code structure of the noise codes if that method of coding was being used. To the best of applicant's knowledge, this is a novel approach to the mature field of steganography.

In all three of the foregoing applications of universal codes, it will often be desirable to append a short (perhaps 8- or 16-bit) private code, which users would keep in their own secured places, in addition to the universal code. This affords the user a further modicum of security against potential erasure of the universal codes by sophisticated pirates.

Applicant's Prior Application

The Detailed Description to this point has simply repeated the disclosure of applicant's prior international application, laid open as PCT publication WO 95/14289. It was reproduced above simply to provide context for the disclosure which follows.

One Master Code Signal As A Distinction From N Independent Embedded Code Signals

In certain sections of this disclosure, perhaps exemplified in the section on the realtime encoder, an economizing step was taken whereby the N independent and source-signal-coextensive embedded code signals were so designed that the non-zero elements of any given embedded code signal were unique to just that embedded code signal and no others. Said more carefully, certain pixels/sample points of a given signal were "assigned" to some pre-determined m'th bit location in our N-bit identification word. Furthermore, and as another basic optimization of implementation, the aggregate of these assigned pixels/samples across all N embedded code signals is precisely the extent of the source signal, meaning each and every pixel/sample location in a source signal is assigned one and only one m'th bit place in our N-bit identification word. (This is not to say, however, that each and every pixel MUST be modified). As a matter of simplification we can then talk about a single master code signal (or "Snowy Image") rather than N independent signals, realizing that pre-defined locations in this master signal correspond to unique bit locations in our N-bit identification word. We therefore construct, via this circuitous route, this rather simple concept on the single master noise signal. Beyond mere economization and simplification, there are also performance reasons for this shift, primarily derived from the idea that individual bit places in our N-bit identification word are no longer "competing" for the information carrying capacity of a single pixel/sample.

With this single master more clearly understood, we can gain new insights into other sections of this disclosure and explore further details within the given application areas.

More of Deterministic Universal Codes Using the Master Code Concept

One case in point is to further explore the use of Deterministic Universal Codes, labelled as item "2" in the sections devoted to universal codes. A given user of this technology may opt for the following variant use of the principles of this technology. The user in question might be a mass distributor of home videos, but clearly the principles would extend to all other potential users of this technology. Fig. 13 pictorially represents the steps involved. In the example the user is one "Alien Productions." They first create an image canvas which is coextensive to the size of the video frames of their movie "Bud's Adventures." On this canvas they print the name of the movie, they place their logo and company name. Furthermore, they have specific information at the bottom, such as the distribution lot for the mass copying that they are currently cranking out, and as indicated, they actually have a unique frame number indicated. Thus we find the example of a standard image 700 which forms the initial basis for the creation of a master Snowy Image (master code signal) which will be added into the original movie frame, creating an output distributable frame. This image 700 can be either black & white or color. The process of turning this image 700 into a pseudo random master code signal is alluded to by the encryption/scrambling routine 702, wherein the original image 700 is passed through any of dozens of well known scrambling methods. The depiction of the number "28" alludes to the idea that there can actually be a library of scrambling methods, and the particular method used for this particular movie, or even for this particular frame, can change. The result is our classic master code signal or Snowy Image. In general, its brightness values are large and it would look very much like the snowy image on a television set tuned to a blank channel, but clearly it has been derived from an informative image 700, transformed through a scrambling 702. (Note: the splotchiness of the example picture is actually a rather poor depiction; it was a function of the crude tools available to the inventor).

This Master Snowy Image 704 is then the signal which is modulated by our N-bit identification word as outlined in other sections of the disclosure, the resulting modulated signal is then scaled down in brightness to the acceptable perceived noise level, and then added to the original frame to produce the distributable frame.

There are a variety of advantages and features that the method depicted in Fig. 13 affords. There are also variations of theme within this overall variation. Clearly, one advantage is that users can now use more intuitive and personalized methods for stamping and signing their work. Provided that the encryption/scrambling routines, 702, are indeed of a high security and not published or leaked, then even if a would-be pirate has knowledge of the logo image 700, they should not be able to use this knowledge to be able to sleuth the Master Snowy Image 704, and thus they should not be able to crack the system, as it were. On the other hand, simple encryption routines 702 may open the door for cracking the system. Another clear advantage of the method of Fig. 13 is the ability to place further information into the overall protective process. Strictly speaking, the information contained in the logo image 700 is not directly

carried in the final distributable frame. Said another way, and provided that the encryption/scrambling routine 702 has a straightforward and known decryption/descrambling method which tolerates bit truncation errors, it is generally impossible to fully re-create the image 700 based upon having the distributable frame, the N-bit identification code word, the brightness scaling factor used, and the number of the decryption routine to be used. The reason that an exact recreation of the image 700 is impossible is due to the scaling operation itself and the concomitant bit truncation. For the present discussion, this whole issue is somewhat academic, however.

A variation on the theme of Fig. 13 is to actually place the N-bit identification code word directly into the logo image 700. In some sense this would be self-referential. Thus when we pull out our stored logo image 700 it already contains visually what our identification word is, then we apply encryption routine #28 to this image, scale it down, then use this version to decode a suspect image using the techniques of this disclosure. The N bit word thus found should match the one contained in our logo image 700.

One desirable feature of the encryption/scrambling routine 702 might be (but is certainly not required to be) that even given a small change in the input image 700, such as a single digit change of the frame number, there would be a huge visual change in the output scrambled master snowy image 704. Likewise, the actual scrambling routine may change as a function of frame numbers, or certain "seed" numbers typically used within pseudo-randomizing functions could change as a function of frame number. All manner of variations are thus possible, all helping to maintain high levels of security. Eventually, engineering optimization considerations will begin to investigate the relationship between some of these randomizing methods, and how they all relate to maintaining acceptable signal strength levels through the process of transforming an uncompressed video stream into a compressed video stream such as with the MPEG compression methodologies.

Another desired feature of the encryption process 702 is that it should be informationally efficient, i.e., that given any random input, it should be able to output an essentially spatially uniform noisy image with little to no residual spatial patterns beyond pure randomness. Any residual correlated patterns will contribute to inefficiency of encoding the N-bit identification word, as well as opening up further tools to would-be pirates to break the system.

Another feature of the method of Fig. 13 is that there is more intuitional appeal to using recognizable symbols as part of a decoding system, which should then translate favorably in the essentially lay environment of a courtroom. It strengthens the simplicity of the coin flip vernacular mentioned elsewhere. Jury members or judges will better relate to an owner's logo as being a piece of the key of recognizing a suspect copy as being a knock-off.

It should also be mentioned that, strictly speaking, the logo image 700 does not need to be randomized. The steps could equally apply straight to the logo image 700 directly. It is not entirely clear

to the inventor what practical goal this might have. A trivial extension of this concept to the case where N = 1 is where, simply and easily, the logo image 700 is merely added to an original image at a very low brightness level. The inventor does not presume this trivial case to be at all a novelty. In many ways this is similar to the age old issue of subliminal advertising, where the low light level patterns added to an image are recognizable to the human eye/brain system and - supposedly - operating on the human brain at an unconscious level. By pointing out these trivial extensions of the current technology, hopefully there can arise further clarity which distinguishes applicant's novel principles in relation to such well known prior art techniques.

5-bit Abridged Alphanumeric Code Sets and Others

It is desirable in some applications for the N-bit identification word to actually signify names, companies, strange words, messages, and the like. Most of this disclosure focuses on using the N-bit identification word merely for high statistical security, indexed tracking codes, and other index based message carrying. The information carrying capacity of "invisible signatures" inside imagery and audio is somewhat limited, however, and thus it would be wise to use our N bits efficiently if we actually want to "spell out" alphanumeric items in the N-bit identification word.

One way to do this is to define, or to use an already existing, reduced bit (e.g. less than 8-bit ASCII) standardized codes for passing alphanumeric messages. This can help to satisfy this need on the part of some applications. For example, a simple alphanumeric code could be built on a 5-bit index table, where for example the letters V,X,Q, and Z are not included, but the digits 0 through 9 are included. In this way, a 100 bit identification word could carry with it 20 alphanumeric symbols. Another alternative is to use variable bit length codes such as the ones used in text compression routines (e.g. Huffman) whereby more frequently used symbols have shorter bit length codes and less frequently used symbols have longer bit lengths.

More on Detecting and Recognizing the N-bit Identification Word in Suspect Signals

Classically speaking, the detection of the N-bit identification word fits nicely into the old art of detecting known signals in noise. Noise in this last statement can be interpreted very broadly, even to the point where an image or audio track itself can be considered noise, relative to the need to detect the underlying signature signals. One of many references to this older art is the book Kassam, Saleem A., "Signal Detection in Non-Gaussian Noise," Springer-Verlag, 1988 (generally available at well stocked libraries, e.g. available at the U.S. Library of Congress by catalog number TK5102.5 .K357 1988.

To the best of this inventor's current understanding, none of the material

in this book is directly applicable to the issue of discovering the polarity of applicant's embedded signals, but the broader principles are indeed applicable.

In particular, section 1.2 "Basic Concepts of Hypothesis Testing" of Kassam's book lays out the basic concept of a binary hypothesis, assigning the value "1" to one hypothesis and the value "0" to the other hypothesis. The last paragraph of that section is also on point regarding the earlier-described embodiment, i.e., that the "0" hypothesis corresponds to "noise only" case, whereas the "1" corresponds to the presence of a signal in the observations. Applicant's use of true polarity is not like this, however, where now the "0" corresponds to the presence of an inverted signal rather than to "noise-only." Also in the present embodiment, the case of "noise-only" is effectively ignored, and that an identification process will either come up with our N-bit identification word or it will come up with "garbage."

The continued and inevitable engineering improvement in the detection of embedded code signals will undoubtedly borrow heavily from this generic field of known signal detection. A common and well-known technique in this field is the so-called "matched filter," which is incidentally discussed early in section 2 of the Kassam book. Many basic texts on signal processing include discussions on this method of signal detection. This is also known in some fields as correlation detection. Furthermore, when the phase or location of a known signal is known a priori, such as is often the case in applications of this technology, then the matched filter can often be reduced to a simple vector dot product between a suspect image and the embedded signal associated with an m'th bit plane in our N-bit identification word. This then represents yet another simple "detection algorithm" for taking a suspect image and producing a sequence of 1s and 0s with the intention of determining if that series corresponds to a pre-embedded N-bit identification word. In words, and with reference to Fig. 3, we run through the process steps up through and including the subtracting of the original image from the suspect, but the next step is merely to step through all N random independent signals and perform a simple vector dot product between these signals and the difference signal, and if that dot product is negative, assign a '0' and if that dot product is positive, assign a '1.' Careful analysis of this "one of many" algorithms will show its similarity to the traditional matched filter.

There are also some immediate improvements to the "matched filter" and "correlation-type" that can provide enhanced ability to properly detect very low level embedded code signals. Some of these improvements are derived from principles set forth in the Kassam book, others are generated by the inventor and the inventor has no knowledge of their being developed in other papers or works, but neither has the inventor done fully extensive searching for advanced signal detection techniques. One such technique is perhaps best exemplified by figure 3.5 in Kassam on page 79, wherein there are certain plots of the various locally optimum weighting coefficients which can apply to a general dot-product algorithmic approach to detection. In other words, rather than performing a simple dot product, each elemental multiplication operation in an overall dot product can be weighted based upon known *a priori* statistical

information about the difference signal itself, i.e., the signal within which the low level known signals are being sought. The interested reader who is not already familiar with these topics is encouraged to read chapter 3 of Kassam to gain a fuller understanding.

One principle which did not seem to be explicitly present in the Kassam book and which was developed rudimentarily by the inventor involves the exploitation of the magnitudes of the statistical properties of the known signal being sought relative to the magnitude of the statistical properties of the suspect signal as a whole. In particular, the problematic case seems to be where the embedded signals we are looking for are of much lower level than the noise and corruption present on a difference signal. Fig. 14 attempts to set the stage for the reasoning behind this approach. The top figure 720 contains a generic look at the differences in the histograms between a typical "problematic" difference signal, i.e., a difference signal which has a much higher overall energy than the embedded signals that may or may not be within it. The term "mean-removed" simply means that the means of both the difference signal and the embedded code signal have been removed, a common operation prior to performing a normalized dot product. The lower figure 722 then has a generally similar histogram plot of the derivatives of the two signals, or in the case of an image, the scalar gradients. From pure inspection it can be seen that a simple thresholding operation in the derivative transform domain, with a subsequent conversion back into the signal domain, will go a long way toward removing certain innate biases on the dot product "recognition algorithm" of a few paragraphs back. Thresholding here refers to the idea that if the absolute value of a difference signal derivative value exceeds some threshold, then it is replaced simply by that threshold value. The threshold value can be so chosen to contain most of the histogram of the embedded signal.

Another operation which can be of minor assistance in "alleviating" some of the bias effects in the dot product algorithm is the removal of the low order frequencies in the difference signal, i.e., running the difference signal through a high pass filter, where the cutoff frequency for the high pass filter is relatively near the origin (or DC) frequency.

Special Considerations for Recognizing Embedded Codes on Signals Which Have Been Compressed and Decompressed, or Alternatively, for Recognizing Embedded Codes Within Any Signal Which Has Undergone Some Known Process Which Creates Non-Uniform Error Sources

Long title for a basic concept. Some signal processing operations, such as compressing and decompressing an image, as with the JPEG/MPEG formats of image/video compression, create errors in some given transform domain which have certain correlations and structure. Using JPEG as an example, if a given image is compressed then decompressed at some high compression ratio, and that resulting image is then fourier transformed and compared to the fourier transform of the original uncompressed image, a definite pattern is clearly visible. This patterning is indicative of correlated error, i.e. error which can be to some extent quantified and predicted. The prediction of the grosser properties of this

correlated error can then be used to advantage in the heretofore-discussed methods of recognizing the embedded code signals within some suspect image which may have undergone either JPEG compression or any other operation which leaves these telltale correlated error signatures. The basic idea is that in areas where there are known higher levels of error, the value of the recognition methods is diminished relative to the areas with known lower levels of correlated errors. It is often possible to quantify the expected levels of error and use this quantification to appropriately weight the retransformed signal values. Using JPEG compression again as an example, a suspect signal can be fourier transformed, and the fourier space representation may clearly show the telltale box grid pattern. The fourier space signal can then be "spatially filtered" near the grid points, and this filtered representation can then be transformed back into its regular time or space domain to then be run through the recognition methods presented in this disclosure. Likewise, any signal processing method which creates non-uniform error sources can be transformed into the domain in which these error sources are non-uniform, the values at the high points of the error sources can be attenuated, and the thusly "filtered" signal can be transformed back into the time/space domain for standard recognition. Often this whole process will include the lengthy and arduous step of "characterizing" the typical correlated error behavior in order to "design" the appropriate filtering profiles.

"SIGNATURE CODES" and "INVISIBLE SIGNATURES"

Briefly and for the sake of clarity, the phrases and terms "signatures," "invisible signatures," and "signature codes" have been and will continue to be used to refer to the general techniques of this technology and often refer specifically to the composite embedded code signal as defined early on in this disclosure.

MORE DETAILS ON EMBEDDING SIGNATURE CODES INTO MOTION PICTURES

Just as there is a distinction made between the JPEG standards for compressing still images and the MPEG standards for compressed motion images, so too should there be distinctions made between placing invisible signatures into still images and placing signatures into motion images. As with the JPEG/MPEG distinction, it is not a matter of different foundations, it is the fact that with motion images a new dimension of engineering optimization opens up by the inclusion of time as a parameter. Any textbook dealing with MPEG will surely contain a section on how MPEG is (generally) not merely applying JPEG on a frame by frame basis. It will be the same with the application of the principles of this technology: generally speaking, the placement of invisible signatures into motion image sequences will not be simply independently placing invisible signatures into one frame after the next. A variety of time-based considerations come into play, some dealing with the psychophysics of motion image perception, others driven by simple cost engineering considerations.

One embodiment actually uses the MPEG compression standard as a piece of a solution. Other motion image compression schemes could equally well be used, be they already invented or yet to be invented. This example also utilizes the scrambled logo image approach to generating the master snowy image as depicted in Fig. 13 and discussed in the disclosure.

A "compressed master snowy image" is independently rendered as depicted in Fig. 15. "Rendered" refers to the generally well known technique in video, movie and animation production whereby an image or sequence of images is created by constructive techniques such as computer instructions or the drawing of animation cells by hand. Thus, "to render" a signature movie in this example is essentially to let either a computer create it as a digital file or to design some custom digital electronic circuitry to create it.

The overall goal of the procedure outlined in Fig. 15 is to apply the invisible signatures to the original movie 762 in such a way that the signatures do not degrade the commercial value of the movie, memorialized by the side-by-side viewing, 768, AND in such a way that the signature optimally survives through the MPEG compression and decompression process. As noted earlier, the use of the MPEG process in particular is an example of the generic process of compression. Also it should be noted that the example presented here has definite room for engineering variations. In particular, those practiced in the art of motion picture compression will appreciate the fact if we start out with two video streams A and B, and we compress A and B separately and combine their results, then the resultant video stream C will not generally be the same as if we pre-added the video streams A and B and compressed this resultant. Thus we have in general, e.g.:

$$MPEG(A) + MPEG(B) \; =\backslash= \; MPEG(A+B)$$

where =\= is not equal to. This is somewhat an abstract notion to introduce at this point in the disclosure and will become more clear as Fig. 15 is discussed. The general idea, however, is that there will be a variety of algebras that can be used to optimize the pass-through of "invisible" signatures through compression procedures. Clearly, the same principles as depicted in Fig. 15 also work on still images and the JPEG or any other still image compression standard.

Turning now to the details of Fig. 15, we begin with the simple stepping through of all Z frames of a movie or video. For a two hour movie played at 30 frames per second, Z turns out to be (30*2*60*60) or 216,000. The inner loop of 700, 702 and 704 merely mimics Fig. 13's steps. The logo frame optionally can change during the stepping through frames. The two arrows emanating from the box 704 represent both the continuation of the loop 750 and the depositing of output frames into the rendered master Snowy Image 752.

To take a brief but potentially appropriate digression at this point, the use of the concept of a Markov process brings certain clarity to the discussion of optimizing the engineering implementation of the methods of Fig. 15. Briefly, a Markov process is one in which a sequence of events takes place and in general there is no memory between one step in the sequence and the next. In the context of Fig. 15 and a sequence of images, a Markovian sequence of images would be one in which there is no apparent or appreciable correlation between a given frame and the next. Imagine taking the set of all movies ever produced, stepping one frame at a time and selecting a random frame from a random movie to be inserted into an output movie, and then stepping through, say, one minute or 1800 of these frames. The resulting "movie" would be a fine example of a Markovian movie. One point of this discussion is that depending on how the logo frames are rendered and depending on how the encryption/scrambling step 702 is performed, the Master Snowy Movie 752 will exhibit some generally quantifiable degree of Markovian characteristics. The point of this point is that the compression procedure itself will be affected by this degree of Markovian nature and thus needs to be accounted for in designing the process of Fig. 15. Likewise, and only in general, even if a fully Markovian movie is created in the High Brightness Master Snowy Movie, 752, then the processing of compressing and decompressing that movie 752, represented as the MPEG box 754, will break down some of the Markovian nature of 752 and create at least a marginally non-Markovian compressed master Snowy Movie 756. This point will be utilized when the disclosure briefly discusses the idea of using multiple frames of a video stream in order to find a single N-bit identification word, that is, the same N-bit identification word may be embedded into several frames of a movie, and it is quite reasonable to use the information derived from those multiple frames to find that single N-bit identification word. The non-Markovian nature of 756 thus adds certain tools to reading and recognizing the invisible signatures. Enough of this tangent.

With the intent of pre-conditioning the ultimately utilized Master Snowy Movie 756, we now send the rendered High Brightness Master Snowy Movie 752 through both the MPEG compression AND decompression procedure 754. With the caveat previously discussed where it is acknowledged that the MPEG compression process is generally not distributive, the idea of the step 754 is to crudely segregate the initially rendered Snowy Movie 752 into two components, the component which survives the compression process 754 which is 756, and the component which does not survive, also crudely estimated using the difference operation 758 to produce the "Cheap Master Snowy Movie" 760. The reason use is made of the deliberately loose term "Cheap" is that we can later add this signature signal as well to a distributable movie, knowing that it probably won't survive common compression processes but that nevertheless it can provide "cheap" extra signature signal energy for applications or situations which will never experience compression. [Thus it is at least noted in Fig. 15]. Back to Fig. 15 proper, we now have a rough cut at signatures which we know have a higher likelihood of surviving intact through the compression process, and we use this "Compressed Master Snowy Movie" 756 to then go through this

procedure of being scaled down 764, added to the original movie 766, producing a candidate distributable movie 770, then compared to the original movie (768) to ensure that it meets whatever commercially viable criteria which have been set up (i.e. the acceptable perceived noise level). The arrow from the side-by-side step 768 back to the scale down step 764 corresponds quite directly to the "experiment visually..." step of Fig. 2, and the gain control 226 of Fig. 6. Those practiced in the art of image and audio information theory can recognize that the whole of Fig. 15 can be summarized as attempting to pre-condition the invisible signature signals in such a way that they are better able to withstand even quite appreciable compression. To reiterate a previously mentioned item as well, this idea equally applies to ANY such pre-identifiable process to which an image, and image sequence, or audio track might be subjected. This clearly includes the JPEG process on still images.

Additional Elements of the Realtime Encoder Circuitry

It should be noted that the method steps represented in Fig. 15, generally following from box 750 up through the creation of the compressed master snowy movie 756, could with certain modification be implemented in hardware. In particular, the overall analog noise source 206 in Fig. 6 could be replaced by such a hardware circuit. Likewise the steps and associated procedures depicted in Fig. 13 could be implemented in hardware and replace the analog noise source 206.

Recognition based on more than one frame: non-Markovian signatures

As noted in the digression on Markov and non-Markov sequences of images, it is pointed out once again that in such circumstances where the embedded invisible signature signals are non-Markovian in nature, i.e., that there is some correlation between the master snowy image of one frame to that of the next, AND furthermore that a single N-bit identification word is used across a range of frames and that the sequence of N-bit identification words associated with the sequence of frames is not Markovian in nature, then it is possible to utilize the data from several frames of a movie or video in order to recognize a single N-bit identification word. All of this is a fancy way of saying that the process of recognizing the invisible signatures should use as much information as is available, in this case translating to multiple frames of a motion image sequence.

HEADER VERIFICATION

The concept of the "header" on a digital image or audio file is a well established practice in the art. The top of Fig. 16 has a simplified look at the concept of the header, wherein a data file begins with generally a comprehensive set of information about the file as a whole, often including information about who the author or copyright holder of the data is, if there is a copyright holder at all. This header 800 is

then typically followed by the data itself 802, such as an audio stream, a digital image, a video stream, or compressed versions of any of these items. This is all exceedingly known and common in the industry.

One way in which the principles of this technology can be employed in the service of information integrity is generically depicted in the lower diagram of Fig. 16. In general, the N-bit identification word can be used to essentially "wallpaper" a given simple message throughout an image (as depicted) or audio data stream, thereby reinforcing some message already contained in a traditional header. This is referred to as "header verification" in the title of this section. The thinking here is that less sophisticated would-be pirates and abusers can alter the information content of header information, and the more secure techniques of this technology can thus be used as checks on the veracity of header information. Provided that the code message, such as "joe's image" in the header, matches the repeated message throughout an image, then a user obtaining the image can have some higher degree of confidence that no alteration of the header has taken place.

Likewise, the header can actually carry the N-bit identification word so that the fact that a given data set has been coded via the methods of this technology can be highlighted and the verification code built right into the header. Naturally, this data file format has not been created yet since the principles of this technology are currently not being employed.

THE "BODIER": THE ABILITY TO LARGELY REPLACE A HEADER

Although all of the possible applications of the following aspect of applicant's technology are not fully developed, it is nevertheless presented as a design alternative that may be important some day. The title of this section contains the silly phrase used to describe this possibility: the "bodier."

Whereas the previous section outlined how the N-bit identification word could "verify" information contained within the header of a digital file, there is also the prospect that these methods could completely replace the very concept of the header and place the information which is traditionally stored in the header directly into the digital signal and empirical data itself.

This could be as simple as standardizing on, purely for example, a 96-bit (12 bytes) leader string on an otherwise entirely empirical data stream. This leader string would plain and simple contain the numeric length, in elemental data units, of the entire data file not including the leader string, and the number of bits of depth of a single data element (e.g. its number of grey levels or the number of discrete signal levels of an audio signal). From there, universal codes as described in this specification would be used to read the N-bit identification word written directly within the empirical data. The length of the empirical data would need to be long enough to contain the full N bits. The N-bit word would effectively transmit what would otherwise be contained in a traditional header.

Fig. 17 depicts such a data format and calls it the "universal empirical data format." The leader string 820 is comprised of the 64 bit string length 822 and the 32 bit data word size 824. The data stream

826 then immediately follows, and the information traditionally contained in the header but now contained directly in the data stream is represented as the attached dotted line 828. Another term used for this attached information is a "shadow channel" as also depicted in Fig. 17.

Yet another element that may need to be included in the leader string is some sort of complex check sum bits which can verify that the whole of the data file is intact and unaltered. This is not included in Fig. 17.

MORE ON DISTRIBUTED UNIVERSAL CODE SYSTEMS: DYNAMIC CODES

One intriguing variation on the theme of universal codes is the possibility of the N-bit identification word actually containing instructions which vary the operations of the universal code system itself. One of many examples is immediately in order: a data transmission is begun wherein a given block of audio data is fully transmitted, an N-bit identification word is read knowing that the first block of data used universal codes #145 out of a set of 500, say, and that part of the N-bit identification word thus found is the instructions that the next block of data should be "analyzed" using the universal code set #411 rather than #145. In general, this technology can thus be used as a method for changing on the fly the actual decoding instructions themselves. Also in general, this ability to utilize "dynamic codes" should greatly increase the sophistication level of the data verification procedures and increase the economic viability of systems which are prone to less sophisticated thwarting by hackers and would-be pirates. The inventor does not believe that the concept of dynamically changing decoding/decrypting instructions is novel *per se*, but the carrying of those instructions on the "shadow channel" of empirical data does appear to be novel to the best of the inventor's understanding. [Shadow channel was previously defined as yet another vernacular phrase encapsulating the more steganographic proper elements of this technology].

A variant on the theme of dynamic codes is the use of universal codes on systems which have *a priori* assigned knowledge of which codes to use when. One way to summarize this possibility is the idea of "the daily password." The password in this example represents knowledge of which set of universal codes is currently operative, and these change depending on some set of application-specific circumstances. Presumably many applications would be continually updating the universal codes to ones which had never before been used, which is often the case with the traditional concept of the daily password. Part of a currently transmitted N-bit identification word could be the passing on of the next day's password, for example. Though time might be the most common trigger events for the changing of passwords, there could be event based triggers as well.

SYMMETRIC PATTERNS AND NOISE PATTERNS: TOWARD A ROBUST UNIVERSAL CODING SYSTEM

The placement of identification patterns into images is certainly not new. Logos stamped into corners of images, subtle patterns such as true signatures or the wallpapering of the copyright circle-C symbol, and the watermark proper are all examples of placing patterns into images in order to signify ownership or to try to prevent illicit uses of the creative material.

What does appear to be novel is the approach of placing independent "carrier" patterns, which themselves are capable of being modulated with certain information, directly into images and audio for the purposes of transmission and discernment of said information, while effectively being imperceptible and/or unintelligible to a perceiving human. Steganographic solutions currently known to the inventor all place this information "directly" into empirical data (possibly first encrypted, then directly), whereas the methods of this disclosure posit the creation of these (most-often) coextensive carrier signals, the modulation of those carrier signals with the information proper, THEN the direct application to the empirical data.

In extending these concepts one step further into the application arena of universal code systems, where a sending site transmits empirical data with a certain universal coding scheme employed and a receiving site analyzes said empirical data using the universal coding scheme, it would be advantageous to take a closer look at the engineering considerations of such a system designed for the transmission of images or motion images, as opposed to audio. Said more clearly, the same type of analysis of a specific implementation such as is contained in Fig. 9 and its accompanying discussion on the universal codes in audio applications should as well be done on imagery (or two dimensional signals). This section is such an analysis and outline of a specific implementation of universal codes and it attempts to anticipate various hurdles that such a method should clear.

The unifying theme of one implementation of a universal coding system for imagery and motion imagery is "symmetry." The idea driving this couldn't be more simple: a prophylactic against the use of image rotation as a means for less sophisticated pirates to bypass any given universal coding system. The guiding principle is that the universal coding system should easily be read no matter what rotational orientation the subject imagery is in. These issues are quite common in the fields of optical character recognition and object recognition, and these fields should be consulted for further tools and tricks in furthering the engineering implementation of this technology. As usual, an immediate example is in order.

Digital Video And Internet Company XYZ has developed a delivery system of its product which relies on a non-symmetric universal coding which double checks incoming video to see if the individual frames of video itself, the visual data, contain XYZ's own relatively high security internal signature codes using the methods of this technology. This works well and fine for many delivery situations, including their Internet tollgate which does not pass any material unless both the header information is verified AND the in-frame universal codes are found. However, another piece of their commercial network performs mundane routine monitoring on Internet channels to look for unauthorized transmission of their proprietary creative property. They control the encryption procedures used, thus it is no problem for them to

unencrypt creative property, including headers, and perform straightforward checks. A pirate group that wants to traffic material on XYZ's network has determined how to modify the security features in XYZ's header information system, and they have furthermore discovered that by simply rotating imagery by 10 or 20 degrees, and transmitting it over XYZ's network, the network doesn't recognize the codes and therefore does not flag illicit uses of their material, and the receiver of the pirate's rotated material simply unrotates it.

Summarizing this last example via logical categories, the non-symmetric universal codes are quite acceptable for the "enablement of authorized action based on the finding of the codes," whereas it can be somewhat easily by-passed in the case of "random monitoring (policing) for the presence of codes." [Bear in mind that the non-symmetric universal codes may very well catch 90% of illicit uses, i.e. 90% of the illicit users wouldn't bother even going to the simple by-pass of rotation.] To address this latter category, the use of quasi-rotationally symmetric universal codes is called for. "Quasi" derives from the age old squaring the circle issue, in this instance translating into not quite being able to represent a full incrementally rotational symmetric 2-D object on a square grid of pixels. Furthermore, basic considerations must be made for scale/magnification changes of the universal codes. It is understood that the monitoring process must be performed when the monitored visual material is in the "perceptual" domain, i.e. when it has been unencrypted or unscrambled and in the form with which it is (or would be) presented to a human viewer. Would-be pirates could attempt to use other simple visual scrambling and unscrambling techniques, and tools could be developed to monitor for these telltale scrambled signals. Said another way, would-be pirates would then look to transform visual material out of the perceptual domain, pass by a monitoring point, and then transform the material back into the perceptual domain; tools other than the monitoring for universal codes would need to be used in such scenarios. The monitoring discussed here therefore applies to applications where monitoring can be performed in the perceptual domain, such as when it is actually sent to viewing equipment.

The "ring" is the only full rotationally symmetric two dimensional object. The "disk" can be seen as a simple finite series of concentric and perfectly abutted rings having width along their radial axis. Thus, the "ring" needs to be the starting point from which a more robust universal code standard for images is found. The ring also will fit nicely into the issue of scale/magnification changes, where the radius of a ring is a single parameter to keep track of and account for. Another property of the ring is that even the case where differential scale changes are made to different spatial axes in an image, and the ring turns into an oval, many of the smooth and quasi-symmetric properties that any automated monitoring system will be looking for are generally maintained. Likewise, appreciable geometric distortion of any image will clearly distort rings but they can still maintain gross symmetric properties. Hopefully, more pedestrian methods such as simply "viewing" imagery will be able to detect attempted illicit piracy in these regards, especially when such lengths are taken to by-pass the universal coding system.

Rings to Knots

Having discovered the ring as the only ideal symmetric pattern upon whose foundation a full rotationally robust universal coding system can be built, we must turn this basic pattern into something functional, something which can carry information, can be read by computers and other instrumentation, can survive simple transformations and corruptions, and can give rise to reasonably high levels of security (probably not unbreakable, as the section on universal codes explained) in order to keep the economics of subversion as a simple incremental cost item.

One embodiment of the "ring-based" universal codes is what the inventor refers to as "knot patterns" or simply "knots," in deference to woven Celtic knot patterns which were later refined and exalted in the works of Leonardo Da Vinci (e.g. Mona Lisa, or his knot engravings). Some rumors have it that these drawings of knots were indeed steganographic in nature, i.e. conveying messages and signatures; all the more appropriate. Figs. 18 and 19 explore some of the fundamental properties of these knots.

Two simple examples of knot patterns are depicted by the supra-radial knots, 850 and the radial knots 852. The names of these types are based on the central symmetry point of the splayed rings and whether the constituent rings intersect this point, are fully outside it, or in the case of sub-radial knots the central point would be inside a constituent circle. The examples of 850 and 852 clearly show a symmetrical arrangement of 8 rings or circles. "Rings" is the more appropriate term, as discussed above, in that this term explicitly acknowledges the width of the rings along the radial axis of the ring. It is each of the individual rings in the knot patterns 850 and 852 which will be the carrier signal for a single associated bit plane in our N-bit identification word. Thus, the knot patterns 850 and 852 each are an 8-bit carrier of information. Specifically, assuming now that the knot patterns 850 and 852 are luminous rings on a black background, then the "addition" of a luminous ring to an independent source image could represent a "1" and the "subtraction" of a luminous ring from an independent source image could represent a "0." The application of this simple encoding scheme could then be replicated over and over as in Fig. 19 and its mosaic of knot patterns, with the ultimate step of adding a scaled down version of this encoded (modulated) knot mosaic directly and coextensively to the original image, with the resultant being the distributable image which has been encoded via this universal symmetric coding method. It remains to communicate to a decoding system which ring is the least significant bit in our N-bit identification word and which is the most significant. One such method is to make a slightly ascending scale of radii values (of the individual rings) from the LSB to the MSB. Another is to merely make the MSB, say, 10% larger radius than all the others and to pre-assign counterclockwise as the order with which the remaining bits fall out. Yet another is to put some simple hash mark inside one and only one circle. In other words, there are a variety of ways with which the bit order of the rings can be encoded in these knot patterns.

A procedure for, first, checking for the mere existence of these knot patterns and, second, for reading of the N-bit identification word, is as follows. A suspect image is first fourier transformed via the extremely common 2D FFT computer procedure. Assuming that we don't know the exact scale of the knot patterns, i.e., we don't know the radius of an elemental ring of the knot pattern in the units of pixels, and that we don't know the exact rotational state of a knot pattern, we merely inspect (via basic automated pattern recognition methods) the resulting magnitude of the Fourier transform of the original image for telltale ripple patterns (concentric low amplitude sinusoidal rings on top of the spatial frequency profile of a source image). The periodicity of these rings, along with the spacing of the rings, will inform us that the universal knot patterns are or are not likely present, and their scale in pixels. Classical small signal detection methods can be applied to this problem just as they can to the other detection methodologies of this disclosure. Common spatial filtering can then be applied to the fourier transformed suspect image, where the spatial filter to be used would pass all spatial frequencies which are on the crests of the concentric circles and block all other spatial frequencies. The resulting filtered image would be fourier transformed out of the spatial frequency domain back into the image space domain, and almost by visual inspection the inversion or non-inversion of the luminous rings could be detected, along with identification of the MSB or LSB ring, and the (in this case 8) N-bit identification code word could be found. Clearly, a pattern recognition procedure could perform this decoding step as well.

The preceding discussion and the method it describes has certain practical disadvantages and shortcomings which will now be discussed and improved upon. The basic method was presented in a simple-minded fashion in order to communicate the basic principles involved.

Let's enumerate a few of the practical difficulties of the above described universal coding system using the knot patterns. For one (1), the ring patterns are somewhat inefficient in their "covering" of the full image space and in using all of the information carrying capacity of an image extent. Second (2), the ring patterns themselves will almost need to be more visible to the eye if they are applied, say, in a straightforward additive way to an 8-bit black and white image. Next (3), the "8" rings of Fig. 18, 850 and 852, is a rather low number, and moreover, there is a 22 and one half degree rotation which could be applied to the figures which the recognition methods would need to contend with (360 divided by 8 divided by 2). Next (4), strict overlapping of rings would produce highly condensed areas where the added and subtracted brightness could become quite appreciable. Next (5), the 2D FFT routine used in the decoding is notoriously computationally cumbersome as well as some of the pattern recognition methods alluded to. Finally (6), though this heretofore described form of universal coding does not pretend to have ultra-high security in the classical sense of top security communications systems, it would nevertheless be advantageous to add certain security features which would be inexpensive to implement in hardware and software systems which at the same time would increase the cost of would-be pirates attempting to thwart the system, and increase the necessary sophistication level of those pirates, to the point that a would-be

pirate would have to go so far out of their way to thwart the system that willfulness would be easily proven and hopefully subject to stiff criminal liability and penalty (such as the creation and distribution of tools which strip creative property of these knot pattern codes).

All of these items can be addressed and should continue to be refined upon in any engineering implementation of the principles of the technology. This disclosure addresses these items with reference to the following embodiments.

Beginning with item number 3, that there are only 8 rings represented in Fig. 18 is simply remedied by increasing the number of rings. The number of rings that any given application will utilize is clearly a function of the application. The trade-offs include but are not limited to: on the side which argues to limit the number of rings utilized, there will ultimately be more signal energy per ring (per visibility) if there are less rings; the rings will be less crowded so that there discernment via automated recognition methods will be facilitated; and in general since they are less crowded, the full knot pattern can be contained using a smaller overall pixel extent, e.g. a 30 pixel diameter region of image rather than a 100 pixel diameter region. The arguments to increase the number of rings include: the desire to transmit more information, such as ascii information, serial numbers, access codes, allowed use codes and index numbers, history information, etc.; another key advantage of having more rings is that the rotation of the knot pattern back into itself is reduced, thereby allowing the recognition methods to deal with a smaller range of rotation angles (e.g., 64 rings will have a maximum rotational displacement of just under 3 degrees, i.e. maximally dissimilar to its original pattern, where a rotation of about 5 and one half degrees brings the knot pattern back into its initial alignment; the need to distinguish the MSB/LSB and the bit plane order is better seen in this example as well). It is anticipated that most practical applications will choose between 16 and 128 rings, corresponding to $N=16$ to $N=128$ for the choice of the number of bits in the N-bit identification code word. The range of this choice would somewhat correlate to the overall radius, in pixels, allotted to an elemental knot pattern such as 850 or 852.

Addressing the practical difficulty item number 4, that of the condensation of rings patterns at some points in the image and lack of ring patterns in others (which is very similar, but still distinct from, item 1, the inefficient covering), the following improvement can be applied. Fig. 18 shows an example of a key feature of a "knot" (as opposed to a pattern of rings) in that where patterns would supposedly intersect, a virtual third dimension is posited whereby one strand of the knot takes precedence over another strand in some predefined way; see item 854. In the terms of imagery, the brightness or dimness of a given intersection point in the knot patterns would be "assigned" to one and only one strand, even in areas where more than two strands overlap. The idea here is then extended, 864, to how rules about this assignment should be carried out in some rotationally symmetric manner. For example, a rule would be that, travelling clockwise, an incoming strand to a loop would be "behind" an outgoing strand. Clearly there are a multitude of variations which could be applied to these rules, many which would critically

depend on the geometry of the knot patterns chosen. Other issues involved will probably be that the finite width, and moreover the brightness profile of the width along the normal axis to the direction of a strand, will all play a role in the rules of brightness assignment to any given pixel underlying the knot patterns.

A major improvement to the nominal knot pattern system previously described directly addresses practical difficulties (1), the inefficient covering, (2) the unwanted visibility of the rings, and (6) the need for higher levels of security. This improvement also indirectly address item (4) the overlapping issue, which has been discussed in the last paragraph. This major improvement is the following: just prior to the step where the mosaic of the encoded knot patterns is added to an original image to produce a distributable image, the mosaic of encoded knot patterns, 866, is spatially filtered (using common 2D FFT techniques) by a standardized and (generally smoothly) random phase-only spatial filter. It is very important to note that this phase-only filter is itself fully rotationally symmetric within the spatial frequency domain, i.e. its filtering effects are fully rotationally symmetric. The effect of this phase-only filter on an individual luminous ring is to transform it into a smoothly varying pattern of concentric rings, not totally dissimilar to the pattern on water several instances after a pebble is dropped in, only that the wave patterns are somewhat random in the case of this phase-only filter rather than the uniform periodicity of a pebble wave pattern. Fig. 20 attempts to give a rough (i.e. non-greyscale) depiction of these phase-only filtered ring patterns. The top figure of Fig. 20 is a cross section of a typical brightness contour/profile 874 of one of these phase-only filtered ring patterns. Referenced in the figure is the nominal location of the pre-filtered outer ring center, 870. The center of an individual ring, 872, is referenced as the point around which the brightness profile is rotated in order to fully describe the two dimensional brightness distribution of one of these filtered patterns. Yet another rough attempt to communicate the characteristics of the filtered ring is depicted as 876, a crude greyscale image of the filtered ring. This phase-only filtered ring, 876 will can be referred to as a random ripple pattern.

Not depicted in Fig. 20 is the composite effects of phase-only filtering on the knot patterns of Fig. 18, or on the mosaic of knot patterns 866 in Fig. 19. Each of the individual rings in the knot patterns 850 or 852 will give rise to a 2D brightness pattern of the type 876, and together they form a rather complicated brightness pattern. Realizing that the encoding of the rings is done by making it luminous (1) or "anti-luminous" (0), the resulting phase-only filtered knot patterns begin to take on subtle characteristics which no longer make direct sense to the human eye, but which are still readily discernable to a computer especially after the phase-only filtering is inverse filtered reproducing the original rings patterns.

Returning now to Fig. 19, we can imagine that an 8-bit identification word has been encoded on the knot patterns and the knot patterns phase-only filtered. The resulting brightness distribution would be a rich tapestry of overlapping wave patterns which would have a certain beauty, but would not be readily intelligible to the eye/brain. [An exception to this might draw from the lore of the South Pacific Island communities, where it is said that sea travellers have learned the subtle art of reading small and multiply

complex ocean wave patterns, generated by diffracted and reflected ocean waves off of intervening islands, as a primary navigational tool.] For want of a better term, the resulting mosaic of filtered knot patterns (derived from 866) can be called the encoded knot tapestry or just the knot tapestry. Some basic properties of this knot tapestry are that it retains the basic rotational symmetry of its generator mosaic, it is generally unintelligible to the eye/brain, thus raising it a notch on the sophistication level of reverse engineering, it is more efficient at using the available information content of a grid of pixels (more on this in the next section), and if the basic knot concepts 854 and 864 are utilized, it will not give rise to local "hot spots" where the signal level becomes unduly condensed and hence objectionably visible to a viewer.

The basic decoding process previously described would now need the additional step of inverse filtering the phase-only filter used in the encoding process. This inverse filtering is quite well known in the image processing industry. Provided that the scale of the knot patterns are known *a priori*, the inverse filtering is straightforward. If on the other hand the scale of the knot patterns is not known, then an additional step of discovering this scale is in order. One such method of discovering the scale of the knot patterns is to iteratively apply the inverse phase-only filter to variously scaled version of an image being decoded, searching for which scale-version begins to exhibit noticeable knot patterning. A common search algorithm such as the simplex method could be used in order to accurately discover the scale of the patterns. The field of object recognition should also be consulted, under the general topic of unknown-scale object detection.

An additional point about the efficiency with which the knot tapestry covers the image pixel grid is in order. Most applications of the knot tapestry method of universal image coding will posit the application of the fully encoded tapestry (i.e. the tapestry which has the N-bit identification word embedded) at a relative low brightness level into the source image. In real terms, the brightness scale of the encoded tapestry will vary from, for example, -5 grey scale values to 5 grey scale values in a typical 256 grey scale image, where the preponderance of values will be within -2 and 2. This brings up the purely practical matter that the knot tapestry will be subject to appreciable bit truncation error. Put as an example, imagine a constructed knot tapestry nicely utilizing a full 256 grey level image, then scaling this down by a factor of 20 in brightness including the bit truncation step, then rescaling this truncated version back up in brightness by the same factor of 20, then inverse phase-only filtering the resultant. The resulting knot pattern mosaic will be a noticeably degraded version of the original knot pattern mosaic. The point of bringing all of this up is the following: it will be a simply defined, but indeed challenging, engineering task to select the various free parameters of design in the implementation of the knot tapestry method, the end goal being to pass a maximum amount of information about the N-bit identification word within some pre-defined visibility tolerance of the knot tapestry. The free parameters include but would not be fully limited to: the radius of the elemental ring in pixels, N or the number of rings, the distance in pixels from the center of a knot pattern to the center of an elemental ring, the packing criteria and

distances of one knot pattern with the others, the rules for strand weaving, and the forms and types of phase-only filters to be used on the knot mosaics. It would be desirable to feed such parameters into a computer optimization routine which could assist in their selection. Even this would begin surely as more of an art than a science due to the many non-linear free parameters involved.

A side note on the use of phase-only filtering is that it can assist in the detection of the ring patterns. It does so in that the inverse filtering of the decoding process tends to "blur" the underlying source image upon which the knot tapestry is added, while at the same time "bringing into focus" the ring patterns. Without the blurring of the source image, the emerging ring patterns would have a harder time "competing" with the sharp features of typical images. The decoding procedure should also utilize the gradient thresholding method described in another section. Briefly, this is the method where if it is known that a source signal is much larger in brightness than our signature signals, then an image being decoded can have higher gradient areas thresholded in the service of increasing the signal level of the signature signals relative to the source signal.

As for the other practical difficulty mentioned earlier, item (5) which deals with the relative computational overhead of the 2D FFT routine and of typical pattern recognition routines, the first remedy here posited but not filled is to find a simpler way of quickly recognizing and decoding the polarity of the ring brightnesses than that of using the 2D FFT. Barring this, it can be seen that if the pixel extent of an individual knot pattern (850 or 852) is, for example, 50 pixels in diameter, than a simple 64 by 64 pixel 2D FFT on some section of an image may be more than sufficient to discern the N-bit identification word as previously described. The idea would be to use the smallest image region necessary, as opposed to being required to utilize an entire image, to discern the N-bit identification word.

Another note is that those practitioners in the science of image processing will recognize that instead of beginning the discussion on the knot tapestry with the utilization of rings, we could have instead jumped right to the use of 2D brightness distribution patterns 876, QUA bases functions. The use of the "ring" terminology as the baseline technology is partly didactic, as is appropriate for patent disclosures in any event. What is more important, perhaps, is that the use of true "rings" in the decoding process, post-inverse filtering, is probably the simplest form to input into typical pattern recognition routines.

Neural Network Decoders

Those skilled in the signal processing art will recognize that computers employing neural network architectures are well suited to the pattern recognition and detection-of-small-signal-in-noise issues posed by the present technology. While a complete discourse on these topics is beyond the scope of this specification, the interested reader is referred to, e.g., Cherkassky, V., "From Statistics to Neural Networks: Theory & Pattern Recognition Applications," Springer-Verlag, 1994; Masters, T., "Signal & Image Processing with Neural Networks: C Sourcebook," Wiley, 1994; Guyon, I. "Advances in Pattern

Recognition Systems Using Neural Networks," World Scientific Publishers, 1994; Nigrin, A., "Neural Networks for Pattern Recognition," MIT Press, 1993; Cichoki, A., "Neural Networks for Optimization & Signal Processing," Wiley, 1993; and Chen, C., "Neural Networks for Pattern Recognition & Their Applications," World Scientific Publishers, 1991.

2D UNIVERSAL CODES II : SIMPLE SCAN LINE IMPLEMENTATION OF THE ONE DIMENSIONAL CASE

The above section on rings, knots and tapestries certainly has its beauty, but some of the steps involved may have enough complexity that practical implementations may be too costly for certain applications. A poor cousin the concept of rings and well-designed symmetry is to simply utilize the basic concepts presented in connection with Fig. 9 and the audio signal, and apply them to two dimensional signals such as images, but to do so in a manner where, for example, each scan line in an image has a random starting point on, for example, a 1000 pixel long universal noise signal. It would then be incumbent upon recognition software and hardware to interrogate imagery across the full range of rotational states and scale factors to "find" the existence of these universal codes.

THE UNIVERSAL COMMERCIAL COPYRIGHT (UCC) IMAGE, AUDIO, AND VIDEO FILE FORMATS

It is as well known as it is regretted that there exist a plethora of file format standards (and not-so-standards) for digital images, digital audio, and digital video. These standards have generally been formed within specific industries and applications, and as the usage and exchange of creative digital material proliferated, the various file formats slugged it out in cross-disciplinary arenas, where today we find a *de facto* histogram of devotees and users of the various favorite formats. The JPEG, MPEG standards for formatting and compression are only slight exceptions it would seem, where some concerted cross-industry collaboration came into play.

The cry for a simple universal standard file format for audio/visual data is as old as the hills. The cry for the protection of such material is older still. With all due respect to the innate difficulties attendant upon the creation of a universal format, and with all due respect to the pretentiousness of outlining such a plan within a patent disclosure, the inventor does believe that these methods can serve perhaps as well as anything for being the foundation upon which an accepted world-wide "universal commercial copyright" format is built. Practitioners know that such animals are not built by proclamation, but through the efficient meeting of broad needs, tenacity, and luck. More germane to the purposes of this disclosure is the fact that the application of this technology would benefit if it could become a central piece within an industry standard file format. The use of universal codes in particular could be specified within such a

standard. The fullest expression of the commercial usage of this technology comes from the knowledge that the invisible signing is taking place and the confidence that instills in copyright holders.

The following is a list of reasons that the principles of this technology could serve as the catalyst for such a standard: (1) Few if any technical developments have so isolated and so pointedly addressed the issue of broad-brush protection of empirical data and audio/visual material; (2) All previous file formats have treated the information about the data, and the data itself, as two separate and physically distinct entities, whereas the methods of this technology can combine the two into one physical entity; (3) The mass scale application of the principles of this technology will require substantial standardization work in the first place, including integration with the years-to-come improvements in compression technologies, so the standards infrastructure will exist by default; (4) the growth of multimedia has created a generic class of data called "content," which includes text, images, sound, and graphics, arguing for higher and higher levels of "content standards"; and (5) marrying copyright protection technology and security features directly into a file format standard is long overdue.

Elements of a universal standard would certainly include the mirroring aspects of the header verification methods, where header information is verified by signature codes directly within data. Also, a universal standard would outline how hybrid uses of fully private codes and public codes would commingle. Thus, if the public codes were "stripped" by sophisticated pirates, the private codes would remain intact. A universal standard would specify how invisible signatures would evolve as digital images and audio evolve. Thus, when a given image is created based on several source images, the standard would specify how and when the old signatures would be removed and replaced by new signatures, and if the header would keep track of these evolutions and if the signatures themselves would keep some kind of record.

PIXELS VS. BUMPS

Most of the disclosure focuses on pixels being the basic carriers of the N-bit identification word. The section discussing the use of a single "master code signal" went so far as to essentially "assign" each and every pixel to a unique bit plane in the N-bit identification word.

For many applications, with one exemplar being that of ink based printing at 300 dots per inch resolution, what was once a pixel in a pristine digital image file becomes effectively a blob (e.g. of dithered ink on a piece of paper). Often the isolated information carrying capacity of the original pixel becomes compromised by neighboring pixels spilling over into the geometrically defined space of the original pixel. Those practiced in the art will recognize this as simple spatial filtering and various forms of blurring.

In such circumstances it may be more advantageous to assign a certain highly local group of pixels to a unique bit plane in the N-bit identification word, rather than merely a single pixel. The end goal is

simply to pre-concentrate more of the signature signal energy into the lower frequencies, realizing that most practical implementations quickly strip or mitigate higher frequencies.

A simple-minded approach would be to assign a 2 by 2 block of pixels all to be modulated with the same ultimate signature grey value, rather than modulating a single assigned pixel. A more fancy approach is depicted in Figs. 21A and 21B, where an array of pixel groups is depicted. This is a specific example of a large class of configurations. The idea is that now a certain small region of pixels is associated with a given unique bit plane in the N-bit identification word, and that this grouping actually shares pixels between bit planes (though it doesn't necessary have to share pixels, as in the case of a 2x2 block of pixels above).

Depicted in Figs. 21A and 21B is a 3x3 array of pixels with an example normalized weighting (normalized --> the weights add up to 1). The methods of this technology now operate on this elementary "bump," as a unit, rather than on a single pixel. It can be seen that in this example there is a fourfold decrease in the number of master code values that need to be stored, due to the spreading out of the signature signal. Applications of this "bump approach" to placing in invisible signatures include any application which will experience *a priori* known high amounts of blurring, where proper identification is still desired even after this heavy blurring.

MORE ON THE STEGANOGRAPHIC USES OF THIS TECHNOLOGY

As mentioned in the initial sections of the disclosure, steganography as an art and as a science is a generic prior art to this technology. Putting the shoe on the other foot now, and as already doubtless apparent to the reader who has ventured thus far, the methods of this technology can be used as a novel method for performing steganography. (Indeed, all of the discussion thus far may be regarded as exploring various forms and implementations of steganography.)

In the present section, we shall consider steganography as the need to pass a message from point A to point B, where that message is essentially hidden within generally independent empirical data. As anyone in the industry of telecommunications can attest to, the range of purposes for passing messages is quite broad. Presumably there would be some extra need, beyond pure hobby, to place messages into empirical data and empirical signals, rather than sending those messages via any number of conventional and straightforward channels. Past literature and product propaganda within steganography posits that such an extra need, among others, might be the desire to hide the fact that a message is even being sent. Another possible need is that a conventional communications channel is not available directly or is cost prohibitive, assuming, that is, that a sender of messages can "transmit" their encoded empirical data somehow. This disclosure includes by reference all previous discussions on the myriad uses to which steganography might apply, while adding the following uses which the inventor has not previously seen described.

The first such use is very simple. It is the need to carry messages about the empirical data within which the message is carried. The little joke is that now the media is truly the message, though it would be next to impossible that some previous steganographer hasn't already exploited this joke. Some of the discussion on placing information about the empirical data directly inside that empirical data was already covered in the section on replacing the header and the concept of the "bodier." This section expands upon that section somewhat.

The advantages of placing a message about empirical data directly in that data is that now only one class of data object is present rather than the previous two classes. In any two class system, there is the risk of the two classes becoming disassociated, or one class corrupted without the other knowing about it. A concrete example here is what the inventor refers to as "device independent instructions."

There exist zillions of machine data formats and data file formats. This plethora of formats has been notorious in its power to impede progress toward universal data exchange and having one machine do the same thing that another machine can do. The instructions that an originator might put into a second class of data (say the header) may not at all be compatible with a machine which is intended to recognize these instructions. If format conversions have taken place, it is also possible that critical instructions have been stripped along the way, or garbled. The improvements disclosed here can be used as a way to "seal in" certain instructions directly into empirical data in such a way that all that is needed by a reading machine to recognize instructions and messages is to perform a standardized "recognition algorithm" on the empirical data (providing of course that the machine can at the very least "read" the empirical data properly). All machines could implement this algorithm any old way they choose, using any compilers or internal data formats that they want.

Implementation of this device independent instruction method would generally not be concerned over the issue of piracy or illicit removal of the sealed in messages. Presumably, the embedded messages and instructions would be a central valuable component in the basic value and functioning of the material.

Another example of a kind of steganographic use of the technology is the embedding of universal use codes for the benefit of a user community. The "message" being passed could be simply a registered serial number identifying ownership to users who wish to legitimately use and pay for the empirical information. The serial number could index into a vast registry of creative property, containing the name or names of the owners, pricing information, billing information, and the like. The "message" could also be the clearance of free and public use for some given material. Similar ownership identification and use indexing can be achieved in two class data structure methods such as a header, but the use of the single class system of this technology may offer certain advantages over the two class system in that the single class system does not care about file format conversion, header compatibilities, internal data format issues, header/body archiving issues, and media transformations.

Fully Exact Steganography

Prior art steganographic methods currently known to the inventor generally involve fully deterministic or "exact" prescriptions for passing a message. Another way to say this is that it is a basic assumption that for a given message to be passed correctly in its entirety, the receiver of the information needs to receive the exact digital data file sent by the sender, tolerating no bit errors or "loss" of data. By definition, "lossy" compression and decompression on empirical signals defeat such steganographic methods. (Prior art, such as the previously noted Komatsu work, are the exceptions here.)

The principles of this technology can also be utilized as an exact form of steganography proper. It is suggested that such exact forms of steganography, whether those of prior art or those of this technology, be combined with the relatively recent art of the "digital signature" and/or the DSS (digital signature standard) in such a way that a receiver of a given empirical data file can first verify that not one single bit of information has been altered in the received file, and thus verify that the contained exact steganographic message has not been altered.

The simplest way to use the principles of this technology in an exact steganographic system is to utilize the previously discussed "designed" master noise scheme wherein the master snowy code is not allowed to contain zeros. Both a sender and a receiver of information would need access to BOTH the master snowy code signal AND the original unencoded original signal. The receiver of the encoded signal merely subtracts the original signal giving the difference signal and the techniques of simple polarity checking between the difference signal and the master snowy code signal, data sample to data sample, producing a the passed message a single bit at a time. Presumably data samples with values near the "rails" of the grey value range would be skipped (such as the values 0,1,254 and 255 in 8-bit depth empirical data).

Statistical Steganography

The need for the receiver of a steganographic embedded data file to have access to the original signal can be removed by turning to what the inventor refers to as "statistical steganography." In this approach, the methods of this technology are applied as simple *a priori* rules governing the reading of an empirical data set searching for an embedded message. This method also could make good use of it combination with prior art methods of verifying the integrity of a data file, such as with the DSS. (See, e.g., Walton, "Image Authentication for a Slippery New Age," Dr. Dobb's Journal, April, 1995, p. 18 for methods of verifying the sample-by-sample, bit-by-bit, integrity of a digital image.)

Statistical steganography posits that a sender and receiver both have access to the same master snowy code signal. This signal can be entirely random and securely transmitted to both parties, or generated by a shared and securely transmitted lower order key which generates a larger quasi-random master snowy code signal. It is *a priori* defined that 16 bit chunks of a message will be passed within

contiguous 1024 sample blocks of empirical data, and that the receiver will use dot product decoding methods as outlined in this disclosure. The sender of the information pre-checks that the dot product approach indeed produces the accurate 16 bit values (that is, the sender pre-checks that the cross-talk between the carrier image and the message signal is not such that the dot product operation will produce an unwanted inversion of any of the 16 bits). Some fixed number of 1024 sample blocks are transmitted and the same number times 16 bits of message is therefore transmitted. DSS techniques can be used to verify the integrity of a message when the transmitted data is known to only exist in digital form, whereas internal checksum and error correcting codes can be transmitted in situations where the data may be subject to change and transformation in its transmission. In this latter case, it is best to have longer blocks of samples for any given message content size (such as 10K samples for a 16 bit message chunk, purely as an example).

Continuing for a moment on the topic of error correcting steganography, it will be recognized that many decoding techniques disclosed herein operate on the principle of distinguishing pixels (or bumps) which have been augmented by the encoded data from those that have been diminished by the encoded data. The distinguishing of these positive and negative cases becomes increasingly difficult as the delta values (e.g. the difference between an encoded pixel and the corresponding original pixel) approach zero.

An analogous situation arises in certain modem transmission techniques, wherein an ambiguous middle ground separates the two desired signal states (e.g. +/- 1). Errors deriving from incorrect interpretation of this middle ground are sometimes termed "soft errors." Principles from modem technology, and other technologies where such problems arise, can likewise be applied to mitigation of such errors in the present context.

One approach is to weight the "confidence" of each delta determination. If the pixel (bump) clearly reflects one state or the other (e.g. +/- 1), its "confidence" is said to be high, and it is given a proportionately greater weighting. Conversely, if the pixel (bump) is relatively ambiguous in its interpretation, its confidence is commensurately lower, and it is given a proportionately lesser weighting. By weighting the data from each pixel (bump) in accordance with its confidence value, the effects of soft errors can be greatly reduced.

The foregoing procedure, while theoretically simple, relies on weightings which are best determined empirically. Accordingly, such an approach is not necessarily straightforward.

An alternative approach is to assign confidence values not to interpretations of individual pixels, but rather to determination of bit values -- either from an image excerpt, or across the entire image. In such an arrangement, each decoded message bit is given a confidence value depending on the ambiguity (or not) of the image statistics by which its value was determined.

Such confidence weighting can also be used as a helpful adjunct with other error detecting/correcting schemes. For example, in known error correcting polynomials, the above-detailed weighting parameters can be used to further hone polynomial-based discernment of an error's location.

THE "NOISE" IN VECTOR GRAPHICS AND VERY-LOW-ORDER INDEXED GRAPHICS

The methods of this disclosure generally posit the existence of "empirical signals," which is another way of saying signals which have noise contained within them almost by definition. There are two classes of 2 dimensional graphics which are not generally considered to have noise inherent in them: vector graphics and certain indexed bit-mapped graphics. Vector graphics and vector graphic files are generally files which contain exact instructions for how a computer or printer draws lines, curves and shapes. A change of even one bit value in such a file might change a circle to a square, as a very crude example. In other words, there is generally no "inherent noise" to exploit within these files. Indexed bit-mapped graphics refers to images which are composed of generally a small number of colors or grey values, such as 16 in the early CGA displays on PC computers. Such "very-low-order" bit-mapped images usually display graphics and cartoons, rather than being used in the attempted display of a digital image taken with a camera of the natural world. These types of very-low-order bit-mapped graphics also are generally not considered to contain "noise" in the classic sense of that term. The exception is where indexed graphic files do indeed attempt to depict natural imagery, such as with the GIF (graphic interchange format of Compuserve), where the concept of "noise" is still quite valid and the principles of this technology still quite valid. These latter forms often use dithering (similar to pointillist paintings and color newspaper print) to achieve near lifelike imagery.

This section concerns this class of 2 dimensional graphics which traditionally do not contain "noise." This section takes a brief look at how the principles of this technology can still be applied in some fashion to such creative material.

The easiest way to apply the principles of this technology to these "noiseless" graphics is to convert them into a form which is amenable to the application of the principles of this technology. Many terms have been used in the industry for this conversion, including "ripping" a vector graphic (raster image processing) such that a vector graphic file is converted to a greyscale pixel-based raster image. Programs such as Photoshop by Adobe have such internal tools to convert vector graphic files into RGB or greyscale digital images. Once these files are in such a form, the principles of this technology can be applied in a straightforward manner. Likewise, very-low-indexed bitmaps can be converted to an RGB digital image or an equivalent. In the RGB domain, the signatures can be applied to the three color channels in appropriate ratios, or the RGB image can be simply converted into a greyscale/chroma format such as "Lab" in Adobe's Photoshop software, and the signatures can be applied to the "Lightness channel" therein. Since most of the distribution media, such as videotapes, CD-ROMs, MPEG video, digital images, and print are

all in forms which are amenable to the application of the principles of this technology, this conversion from vector graphic form and very-low-order graphic form is often done in any event.

Another way to apply the principles of this technology to vector graphics and very-low-order bitmapped graphics is to recognize that, indeed, there are certain properties to these inherent graphic formats which - to the eye - appear as noise. The primary example is the borders and contours between where a given line or figure is drawn or not drawn, or exactly where a bit-map changes from green to blue. In most cases, a human viewer of such graphics will be keenly aware of any attempts to "modulate signature signals" via the detailed and methodical changing of the precise contours of a graphic object. Nevertheless, such encoding of the signatures is indeed possible. The distinction between this approach and that disclosed in the bulk of this disclosure is that now the signatures must ultimately derive from what already exists in a given graphic, rather than being purely and separately created and added into a signal. This disclosure points out the possibilities here nonetheless. The basic idea is to modulate a contour, a touch right or a touch left, a touch up or a touch down, in such a way as to communicate an N-bit identification word. The locations of the changes contours would be contained in a an analogous master noise image, though now the noise would be a record of random spatial shifts one direction or another, perpendicular to a given contour. Bit values of the N-bit identification word would be encoded, and read, using the same polarity checking method between the applied change and the change recorded in the master noise image.

PLASTIC CREDIT AND DEBIT CARD SYSTEMS BASED ON THE PRINCIPLES OF THE TECHNOLOGY

Growth in the use of plastic credit cards, and more recently debit cards and ATM cash cards, needs little introduction. Nor does there need to be much discussion here about the long history of fraud and illicit uses of these financial instruments. The development of the credit card hologram, and its subsequent forgery development, nicely serves as a historic example of the give and take of plastic card security measures and fraudulent countermeasures. This section will concern itself with how the principles of this technology can be realized in an alternative, highly fraud-proof yet cost effective plastic card-based financial network.

A basic list of desired features for an ubiquitous plastic economy might be as follows: 1) A given plastic financial card is completely impossible to forge; 2) An attempted forged card (a "look-alike") cannot even function within a transaction setting; 3) Intercepted electronic transactions by a would-be thief would not in any way be useful or re-useable; 4) In the event of physical theft of an actual valid card, there are still formidable obstacles to a thief using that card; and 5) The overall economic cost of implementation of the financial card network is equal to or less than that of the current international credit card networks, i.e., the fully loaded cost per transaction is equal to or less than the current norm, allowing

for higher profit margins to the implementors of the networks. Apart from item 5, which would require a detailed analysis of the engineering and social issues involved with an all out implementation strategy, the following use of the principles of this technology may well achieve the above list, even item 5.

Figs. 22 through 26, along with the ensuing written material, collectively outline what is referred to in Fig. 26 as "The Negligible-Fraud Cash Card System." The reason that the fraud-prevention aspects of the system are highlighted in the title is that fraud, and the concomitant lost revenue therefrom, is apparently a central problem in today's plastic card based economies. The differential advantages and disadvantages of this system relative to current systems will be discussed after an illustrative embodiment is presented.

Fig. 22 illustrates the basic unforgeable plastic card which is quite unique to each and every user. A digital image 940 is taken of the user of the card. A computer, which is hooked into the central accounting network, 980, depicted in Fig. 26, receives the digital image 940, and after processing it (as will be described surrounding Fig. 24) produces a final rendered image which is then printed out onto the personal cash card 950. Also depicted in Fig. 22 is a straightforward identification marking, in this case a bar code 952, and optional position fiducials which may assist in simplifying the scanning tolerances on the Reader 958 depicted in Fig. 23.

The short story is that the personal cash card 950 actually contains a very large amount of information unique to that particular card. There are no magnetic strips involved, though the same principles can certainly be applied to magnetic strips, such as an implanted magnetic noise signal (see earlier discussion on the "fingerprinting" of magnetic strips in credit cards; here, the fingerprinting would be prominent and proactive as opposed to passive). In any event, the unique information within the image on the personal cash card 950 is stored along with the basic account information in a central accounting network, 980, Fig. 26. The basis for unbreakable security is that during transactions, the central network need only query a small fraction of the total information contained on the card, and never needs to query the same precise information on any two transactions. Hundreds if not thousands or even tens of thousands of unique and secure "transaction tokens" are contained within a single personal cash card. Would-be pirates who went so far as to pick off transmissions of either encrypted or even unencrypted transactions would find the information useless thereafter. This is in marked distinction to systems which have a single complex and complete "key" (generally encrypted) which needs to be accessed, in its entirety, over and over again. The personal cash card on the other hand contains thousands of separate and secure keys which can be used once, within milliseconds of time, then forever thrown away (as it were). The central network 980 keeps track of the keys and knows which have been used and which haven't.

Fig. 23 depicts what a typical point-of-sale reading device, 958, might look like. Clearly, such a device would need to be manufacturable at costs well in line with, or cheaper than, current cash register

systems, ATM systems, and credit card swipers. Not depicted in Fig. 23 are the innards of the optical scanning, image processing, and data communications components, which would simply follow normal engineering design methods carrying out the functions that are to be described henceforth and are well within the capability of artisans in these fields. The reader 958 has a numeric punch pad 962 on it, showing that a normal personal identification number system can be combined with the overall design of this system adding one more conventional layer of security (generally after a theft of the physical card has occurred). It should also be pointed out that the use of the picture of the user is another strong (and increasingly common) security feature intended to thwart after-theft and illicit use. Functional elements such as the optical window, 960, are shown, mimicking the shape of the card, doubling as a centering mechanism for the scanning. Also shown is the data line cable 966 presumably connected either to a proprietor's central merchant computer system or possibly directly to the central network 980. Such a reader may also be attached directly to a cash register which performs the usual tallying of purchased items. Perhaps overkill on security would be the construction of the reader, 958, as a type of Faraday cage such that no electronic signals, such as the raw scan of the card, can emanate from the unit. The reader 958 does need to contain, preferably, digital signal processing units which will assist in swiftly calculating the dot product operations described henceforth. It also should contain local read-only memory which stores a multitude of spatial patterns (the orthogonal patterns) which will be utilized in the "recognition" steps outlined in Fig. 25 and its discussion. As related in Fig. 23, a consumer using the plastic card merely places their card on the window to pay for a transaction. A user could choose for themselves if they want to use a PIN number or not. Approval of the purchase would presumably happen within seconds, provided that the signal processing steps of Fig. 25 are properly implemented with effectively parallel digital processing hardware.

Fig. 24 takes a brief look at one way to process the raw digital image, 940, of a user into an image with more useful information content and uniqueness. It should be clearly pointed out that the raw digital image itself could in fact be used in the following methods, but that placing in additional orthogonal patterns into the image can significantly increase the overall system. (Orthogonal means that, if a given pattern is multiplied by another orthogonal pattern, the resulting number is zero, where "multiplication of patterns" is meant in the sense of vector dot products; these are all familiar terms and concepts in the art of digital image processing). Fig. 24 shows that the computer 942 can, after interrogating the raw image 970, generate a master snowy image 972 which can be added to the raw image 970 to produce a yet-more unique image which is the image that is printed onto the actual personal cash card, 950. The overall effect on the image is to "texturize" the image. In the case of a cash card, invisibility of the master snowy pattern is not as much of a requirement as with commercial imagery, and one of the only criteria for keeping the master snowy image somewhat lighter is to not obscure the image of the user. The central network, 980, stores the final processed image in the record of the account of the user, and it is this

unique and securely kept image which is the carrier of the highly secure "throw-away transaction keys." This image will therefore be "made available" to all duly connected point-of-sale locations in the overall network. As will be seen, none of the point-of-sale locations ever has knowledge of this image, they merely answer queries from the central network.

Fig. 25 steps through a typical transaction sequence. The figure is laid out via indentations, where the first column are steps performed by the point-of-sale reading device 958, the second column has information transmission steps communicated over the data line 966, and the third column has steps taken by the central network 980 which has the secured information about the user's account and the user's unique personal cash card 950. Though there is some parallelism possible in the implementation of the steps, as is normally practiced in the engineering implementation of such systems, the steps are nevertheless laid out according to a general linear sequence of events.

Step one of Fig. 25 is the standard "scanning" of a personal cash card 950 within the optical window 960. This can be performed using linear optical sensors which scan the window, or via a two dimensional optical detector array such as a CCD. The resulting scan is digitized into a grey scale image and stored in an image frame memory buffer such as a "framegrabber," as is now common in the designs of optical imaging systems. Once the card is scanned, a first image processing step would probably be locating the four fiducial center points, 954, and using these four points to guide all further image processing operations (i.e. the four fiducials "register" the corresponding patterns and barcodes on the personal cash card). Next, the barcode ID number would be extracted using common barcode reading image processing methods. Generally, the user's account number would be determined in this step.

Step two of Fig. 25 is the optional typing in of the PIN number. Presumably most users would opt to have this feature, except those users who have a hard time remembering such things and who are convinced that no one will ever steal their cash card.

Step three of Fig. 25 entails connecting through a data line to the central accounting network and doing the usual communications handshaking as is common in modem-based communications systems. A more sophisticated embodiment of this system would obviate the need for standard phone lines, such as the use of optical fiber data links, but for now we can assume it is a garden variety belltone phone line and that the reader 958 hasn't forgotten the phone number of the central network.

After basic communications are established, step four shows that the point-of-sale location transmits the ID number found in step 1, along with probably an encrypted version of the PIN number (for added security, such as using the ever more ubiquitous RSA encryption methods), and appends the basic information on the merchant who operates the point-of-sale reader 958, and the amount of the requested transaction in monetary units.

Step five has the central network reading the ID number, routing the information accordingly to the actual memory location of that user's account, thereafter verifying the PIN number and checking that

the account balance is sufficient to cover the transaction. Along the way, the central network also accesses the merchant's account, checks that it is valid, and readies it for an anticipated credit.

Step six begins with the assumption that step five passed all counts. If step five didn't, the exit step of sending a NOT OK back to the merchant is not depicted. So, if everything checks out, the central network generates twenty four sets of sixteen numbers, where all numbers are mutually exclusive, and in general, there will be a large but quite definitely finite range of numbers to choose from. Fig. 25 posits the range being 64K or 65536 numbers. It can be any practical number, actually. Thus, set one of the twenty four sets might have the numbers 23199, 54142, 11007, 2854, 61932, 32879, 38128, 48107, 65192, 522, 55723, 27833, 19284, 39970, 19307, and 41090, for example. The next set would be similarly random, but the numbers of set one would be off limits now, and so on through the twenty four sets. Thus, the central network would send (16x24x2 bytes) of numbers or 768 bytes. The actual amount of numbers can be determined by engineering optimization of security versus transmission speed issues. These random numbers are actually indexes to a set of 64K universally *a priori* defined orthogonal patterns which are well known to both the central network and are permanently stored in memory in all of the point-of-sale readers. As will be seen, a would-be thief's knowledge of these patterns is of no use.

Step seven then transmits the basic "OK to proceed" message to the reader, 958, and also sends the 24 sets of 16 random index numbers.

Step eight has the reader receiving and storing all these numbers. Then the reader, using its local microprocessor and custom designed high speed digital signal processing circuitry, steps through all twenty four sets of numbers with the intention of deriving 24 distinct floating point numbers which it will send back to the central network as a "one time key" against which the central network will check the veracity of the card's image. The reader does this by first adding together the sixteen patterns indexed by the sixteen random numbers of a given set, and then performing a common dot product operation between the resulting composite pattern and the scanned image of the card. The dot product generates a single number (which for simplicity we can call a floating point number). The reader steps through all twenty four sets in like fashion, generating a unique string of twenty four floating point numbers.

Step nine then has the reader transmitting these results back to the central network.

Step ten then has the central network performing a check on these returned twenty four numbers, presumably doing its own exact same calculations on the stored image of the card that the central network has in its own memory. The numbers sent by the reader can be "normalized," meaning that the highest absolute value of the collective twenty four dot products can divided by itself (its unsigned value), so that brightness scale issues are removed. The resulting match between the returned values and the central network's calculated values will either be well within given tolerances if the card is valid, and way off if the card is a phony or if the card is a crude reproduction.

Step eleven then has the central network sending word whether or not the transaction was OK, and letting the customer know that they can go home with their purchased goods.

Step twelve then explicitly shows how the merchant's account is credited with the transaction amount.

As already stated, the primary advantage of this plastic card is to significantly reduce fraud, which apparently is a large cost to current systems. This system reduces the possibility of fraud only to those cases where the physical card is either stolen or very carefully copied. In both of these cases, there still remains the PIN security and the user picture security (a known higher security than low wage clerks analyzing signatures). Attempts to copy the card must be performed through "temporary theft" of the card, and require photo-quality copying devices, not simple magnetic card swipers. The system is founded upon a modern 24 hour highly linked data network. Illicit monitoring of transactions does the monitoring party no use whether the transmissions are encrypted or not.

It will be appreciated that the foregoing approach to increasing the security of transactions involving credit and debit card systems is readily extended to any photograph-based identification system. Moreover, the principles of the present technology may be applied to detect alteration of photo ID documents, and to generally enhance the confidence and security of such systems. In this regard, reference is made to Fig. 28, which depicts a photo-ID card or document 1000 which may be, for example, a passport, visa, permanent resident card ("green card"), driver's license, credit card, government employee identification, or a private industry identification badge. For convenience, such photograph-based identification documents will be collectively referred to as photo ID documents.

The photo ID document includes a photograph 1010 that is attached to the document 1000. Printed, human-readable information 1012 is incorporated in the document 1000, adjacent to the photograph 1010. Machine readable information, such as that known as "bar code" may also be included adjacent to the photograph.

Generally, the photo ID document is constructed so that tampering with the document (for example, swapping the original photograph with another) should cause noticeable damage to the card. Nevertheless, skilled forgerers are able to either alter existing documents or manufacture fraudulent photo ID documents in a manner that is extremely difficult to detect.

As noted above, the present technology enhances the security associated with the use of photo ID documents by supplementing the photographic image with encoded information (which information may or may not be visually perceptible), thereby facilitating the correlation of the photographic image with other information concerning the person, such as the printed information 1012 appearing on the document 1000.

In one embodiment, the photograph 1010 may be produced from a raw digital image to which is added a master snowy image as described above in connection with Figs. 22-24. The above-described

central network and point-of-sale reading device (which device, in the present embodiment, may be considered as a point-of-entry or point-of-security photo ID reading device), would essentially carry out the same processing as described with that embodiment, including the central network generation of unique numbers to serve as indices to a set of defined orthogonal patterns, the associated dot product operation carried out by the reader, and the comparison with a similar operation carried out by the central network. If the numbers generated from the dot product operation carried out by the reader and the central network match, in this embodiment, the network sends the OK to the reader, indicating a legitimate or unaltered photo ID document.

In another embodiment, the photograph component 1010 of the identification document 1000 may be digitized and processed so that the photographic image that is incorporated into the photo ID document 1000 corresponds to the "distributable signal" as defined above. In this instance, therefore, the photograph includes a composite, embedded code signal, imperceptible to a viewer, but carrying an N-bit identification code. It will be appreciated that the identification code can be extracted from the photo using any of the decoding techniques described above, and employing either universal or custom codes, depending upon the level of security sought.

It will be appreciated that the information encoded into the photograph may correlate to, or be redundant with, the readable information 1012 appearing on the document. Accordingly, such a document could be authenticated by placing the photo ID document on a scanning system, such as would be available at a passport or visa control point. The local computer, which may be provided with the universal code for extracting the identification information, displays the extracted information on the local computer screen so that the operator is able to confirm the correlation between the encoded information and the readable information 1012 carried on the document.

It will be appreciated that the information encoded with the photograph need not necessarily correlate with other information on an identification document. For example, the scanning system may need only to confirm the existence of the identification code so that the user may be provided with a "go" or "no go" indication of whether the photograph has been tampered with. It will also be appreciated that the local computer, using an encrypted digital communications line, could send a packet of information to a central verification facility, which thereafter returns an encrypted "go" or "no go" indication.

In another embodiment, it is contemplated that the identification code embedded in the photograph may be a robust digital image of biometric data, such as a fingerprint of the card bearer, which image, after scanning and display, may be employed for comparison with the actual fingerprint of the bearer in very high security access points where on-the-spot fingerprint recognition systems (or retinal scans, etc.) are employed.

It will be appreciated that the information embedded in the photograph need not be visually hidden or steganographically embedded. For example, the photograph incorporated into the identification card

may be a composite of an image of the individual and one-, or two-dimensional bar codes. The bar code information would be subject to conventional optical scanning techniques (including internal cross checks) so that the information derived from the code may be compared, for example, to the information printed on the identification document.

It is also contemplated that the photographs of ID documents currently in use may be processed so that information correlated to the individual whose image appears in the photograph may be embedded. In this regard, the reader's attention is directed to the foregoing portion of this description entitled "Use in Printing, Paper, Documents, Plastic-Coated Identification Cards, and Other Material Where Global Embedded Codes Can Be Imprinted," wherein there is described numerous approaches to modulation of physical media that may be treated as "signals" amenable to application of the present technology principles.

NETWORK LINKING METHOD USING INFORMATION EMBEDDED IN DATA OBJECTS THAT HAVE INHERENT NOISE

The diagram of Fig. 27 illustrates the aspect of the technology that provides a network linking method using information embedded in data objects that have inherent noise. In one sense, this aspect is a network navigation system and, more broadly, a massively distributed indexing system that embeds addresses and indices directly within data objects themselves. As noted, this aspect is particularly well-adapted for establishing hot links with pages presented on the World Wide Web (WWW). A given data object effectively contains both a graphical representation and embedded URL address.

As in previous embodiments, this embedding is carried out so that the added address information does not affect the core value of the object so far as the creator and audience are concerned. As a consequence of such embedding, only one class of data objects is present rather than the two classes (data object and discrete header file) that are attendant with traditional WWW links. The advantages of reducing a hot-linked data object to a single class were mentioned above, and are elaborated upon below. In one embodiment of the technology, the World Wide Web is used as a pre-existing hot link based network. The common apparatus of this system is networked computers and computer monitors displaying the results of interactions when connected to the web. This embodiment of the technology contemplates steganographically embedding URL or other address-type information directly into images, videos, audio, and other forms of data objects that are presented to a web site visitor, and which have "gray scale" or "continuous tones" or "gradations" and, consequently, inherent noise. As noted above, there are a variety of ways to realize basic steganographic implementations, all of which could be employed in accordance with the present technology.

With particular reference to Fig. 27, images, quasi-continuous tone graphics, multimedia video and audio data are currently the basic building blocks of many sites 1002, 1004 on the World Wide Web.

Such data will be hereafter collectively referred to as creative data files or data objects. For illustrative purposes, a continuous-tone graphic data object 1006 (diamond ring with background) is depicted in Fig. 27.

Web site tools - both those that develop web sites 1008 and those that allow browsing them 1010 - routinely deal with the various file formats in which these data objects are packaged. It is already common to distribute and disseminate these data objects 1006 as widely as possible, often with the hope on a creator's part to sell the products represented by the objects or to advertise creative services (e.g., an exemplary photograph, with an 800 phone number displayed within it, promoting a photographer's skills and service). Using the methods of this technology, individuals and organizations who create and disseminate such data objects can embed an address link that leads right back to their own node on a network, their own site on the WWW.

A user at one site 1004 needs merely to point and click at the displayed object 1006. The software 1010 identifies the object as a hot link object. The software reads the URL address that is embedded within the object and routes the user to the linked web site 1002, just as if the user had used a conventional web link. That linked site 1002 is the home page or network node of the creator of the object 1006, which creator may be a manufacturer. The user at the first site 1004 is then presented with, for example, an order form for purchasing the product represented by the object 1006.

It will be appreciated that the creators of objects 1006 having embedded URL addresses or indices (which objects may be referred to as "hot objects") and the manufacturers hoping to advertise their goods and services can now spread their creative content like dandelion seeds in the wind across the WWW, knowing that embedded within those seeds are links back to their own home page.

It is contemplated that the object 1006 may include a visible icon 1012 (such as the exemplary "HO" abbreviation shown in Fig. 27) incorporated as part of the graphic. The icon or other subtle indicia would apprise the user that the object is a hot object, carrying the embedded URL address or other information that is accessible via the software 1010.

Any human-perceptible indicium (e.g., a short musical tone) can serve the purpose of apprising the user of the hot object. It is contemplated, however, that no such indicium is required. A user's trial-and-error approach to clicking on a data object having no embedded address will merely cause the software to look for, but not find, the URL address.

The automation process inherent in the use of this aspect of the technology is very advantageous. Web software and web site development tools simply need to recognize this new class of embedded hot links (hot objects), operating on them in real time. Conventional hot links can be modified and supplemented simply by "uploading" a hot object into a web site repository, never requiring a web site programmer to do a thing other than basic monitoring of traffic.

A method of implementing the above described functions of the present technology generally involves the steps of (1) creating a set of standards by which URL addresses are steganographically embedded within images, video, audio, and other forms of data objects; and (2) designing web site development tools and web software such that they recognize this new type of data object (the hot object), the tools being designed such that when the objects are presented to a user and that user points and clicks on such an object, the user's software knows how to read or decode the steganographic information and route the user to the decoded URL address.

The foregoing portions of this description detailed a steganographic implementation (see, generally, Fig. 2 and the text associated therewith) that is readily adapted to implement the present technology. In this regard, the otherwise conventional site development tool 1008 is enhanced to include, for example, the capability to encode a bit-mapped image file with an identification code (URL address, for example) according to the present technology. In the present embodiment, it is contemplated that the commercial or transaction based hot objects may be steganographically embedded with URL addresses (or other information) using any of the universal codes described above.

The foregoing portions of this description also detailed a technique for reading or decoding steganographically embedded information (see, generally, Fig. 3 and the text associated therewith) that is readily adapted to implement the present technology. In this regard, the otherwise conventional user software 1010 is enhanced to include, for example, the capability to analyze encoded bit-mapped files and extract the identification code (URL address, for example).

While an illustrative implementation for steganographically embedding information on a data object has been described, one of ordinary skill will appreciate that any one of the multitude of available steganographic techniques may be employed to carry out the function of the present embodiment.

It will be appreciated that the present embodiment provides an immediate and common sense mechanism whereby some of the fundamental building blocks of the WWW, namely images and sound, can also become hot links to other web sites. Also, the programming of such hot objects can become fully automated merely through the distribution and availability of images and audio. No real web site programming is required. The present embodiment provides for the commercial use of the WWW in such a way that non-programmers can easily spread their message merely by creating and distributing creative content (herein, hot objects). As noted, one can also transition web based hot links themselves from a more arcane text based interface to a more natural image based interface.

Encapsulated Hot Link File Format

As noted above, once steganographic methods of hot link navigation take hold, then, as new file formats and transmission protocols develop, more traditional methods of "header-based" information attachment can enhance the basic approach built by a steganographic-based system. One way to begin

extending the steganographic based hot link method back into the more traditional header-based method is to define a new class of file format which could effectively become the standard class used in network navigation systems. It will be seen that objects beyond images, audio and the like can now become "hot objects", including text files, indexed graphic files, computer programs, and the like.

The encapsulated hot link (EHL) file format simply is a small shell placed around a large range of pre-existing file formats. The EHL header information takes only the first N bytes of a file, followed by a full and exact file in any kind of industry standard format. The EHL super-header merely attaches the correct file type, and the URL address or other index information associating that object to other nodes on a network or other databases on a network.

It is possible that the EHL format could be the method by which the steganographic methods are slowly replaced (but probably never completely). The slowness pays homage to the idea that file format standards often take much longer to create, implement, and get everybody to actually use (if at all). Again, the idea is that an EHL-like format and system built around it would bootstrap onto a system setup based on steganographic methods.

Self Extracting Web Objects

Generally speaking, three classes of data can be steganographically embedded in an object: a number (e.g. a serial or identification number, encoded in binary), an alphanumeric message (e.g. a human readable name or telephone number, encoded in ASCII or a reduced bit code), or computer instructions (e.g. JAVA or HTML instructions). The embedded URLs and the like detailed above begin to explore this third class, but a more detailed exposition of the possibilities may be helpful.

Consider a typical web page, shown in Fig. 27A. It may be viewed as including three basic components: images (#1 - #6), text, and layout.

Applicant's technology can be used to consolidate this information into a self-extracting object, and regenerate the web page from this object.

In accordance with this example, Fig. 27B shows the images of the Fig. 27A web page fitted together into a single RGB mosaiced image. A user can perform this operation manually using existing image processing programs, such as Adobe's Photoshop software, or the operation can be automated by a suitable software program.

Between certain of the image tiles in the Fig. 27B mosaic are empty areas (shown by cross-hatching).

This mosaiced image is then steganographically encoded to embed the layout instructions (e.g. HTML) and the web page text therein. In the empty areas the encoding gain can be maximized since there is no image data to corrupt. The encoded, mosaiced image is then JPEG compressed to form a self extracting web page object.

(JPEG compression is used in this example as a *lingua franca* of image representations. Another such candidate is the GIF file format. Such formats are supported by a variety of software tools and languages, making them well suited as "common carriers" of embedded information. Other image representations can of course be used.)

These objects can be exchanged as any other JPEG images. When the JPEG file is opened, a suitably programmed computer can detect the presence of the embedded information and extract the layout data and text. Among other information, the layout data specifies where the images forming the mosaic are to be located in the final web page. The computer can follow the embedded HTML instructions to recreate the original web page, complete with graphics, text, and links to other URLs.

If the self extracting web page object is viewed by a conventional JPEG viewer, it does not self-extract. However, the user will see the logos and artwork associated with the web page (with noise-like "grouting" between certain of the images). Artisans will recognize that this is in stark contrast to viewing of other compressed data objects (e.g. PKZIP files and self extracting text archives) which typically appear totally unintelligible unless fully extracted.

(The foregoing advantages can largely be achieved by placing the web page text and layout instructions in a header file associated with a JPEG-compressed mosaiced image file. However, industry standardization of the header formats needed to make such systems practical appears difficult, if not impossible.)

Palettes of Steganographically Encoded Images

Once web images with embedded URL information become widespread, such web images can be collected into "palettes" and presented to users as high level navigation tools. Navigation is effected by clicking on such images (e.g. logos for different web pages) rather than clicking on textual web page names. A suitably programmed computer can decode the embedded URL information from the selected image, and establish the requested connection.

In one embodiment, self-extraction of the above-described web page objects automatically generates thumbnail images corresponding to the extracted pages (e.g. representative logos), which are then stored in a subdirectory in the computer's file system dedicated to collecting such thumbnails. In each such thumbnail is embedded a URL, such as the URL of the extracted page or the URL of the site from which the self-extracting object was obtained. This subdirectory can be accessed to display a palette of navigational thumbnails for selection by the user.

Specific Example of a Computer System Linked by Using Information in Data Objects

The present invention allows a digital watermark to be embedded directly into photographs, video, computer images, audio, and other forms of creative property. The range of applications is amazingly

wide, and includes things like video and movie content watermarking for proof of ownership and tracking, watermarking music and audio content, even texturing physical materials such as auto parts. As discussed above, this watermark is imperceptible to the eye, and imperceptible to the ear, but a computer analysis can read the watermark and discover a message it carries. The watermark is repeated throughout the property. It is robust, and typically survives multiple generations of copying, modification, printing, scanning, and compression. A watermark may carry a copyright notice, a unique serial number, a Transaction ID, as well as other application specific data.

Conceptually, the example below is made with respect to a watermark embedded by "PictureMarc" which illustrates the idea by carrying a unique creator id, and a set of image attributes. This creator id corresponds to complete contact information available through an on-line service, MarcCentre. Technically, the watermark is 128 bits long and has a data payload of 76 bits, although the invention is not restricted to any particular watermark size. In this example, it is short so that it can be repeated many times through an image, making it possible to find the message even from a portion of the original image. Other approaches can include data such as an image serial number or a Transaction ID in the watermark message. It is also possible to add a person's name or other such data (to a computer, it is all data), but one application is focused on embedding a more structured message. First, to accommodate a text string, there would be a tendency to push for longer messages. Long messages result in not being able to fit very many copies of the message in an image, and thus the watermark becomes less robust. Also, just putting a name in an image is much like the credit line in a magazine: A. Jones tells who the person is, but not how to contact the person. With a watermark system available on line, a user can get complete and up to date contact details from each Creator ID through the on-line service. Accordingly, it is preferable to associate persistent digital information with an image. The core software has been designed to enable working with vendors or large customers who may want to develop specific watermark message types for special applications such as embedding the film speed and exposure in the image, or serializing images in a large collection for internal tracking and management.

One approach to implementing this embodiment involves "PictureMarc", which includes a writer portion to embed watermarks and a reader portion to find them and read their contents. This bundle is integrated into Adobe Photoshop® 4.0 (Adobe Photoshop is a trademark of Adobe Systems, Inc.) as a plugin extension, and from the end-user's perspective is free. A free reader will also be available through a web site. PictureMarc for Photoshop is supported on Apple

Macintosh and Microsoft Windows platforms. PictureMarc supports embedding a Creator ID and image attributes in an image. PictureMarc also has the ability to carry a unique Image ID or even a Transaction ID in the watermark. These formats will support tracking images both to a collection, and to the person and terms that were in place when the image sold. The watermark reader will accept and read all watermark formats.

As an example, the first step is to read an image from a CD-ROM, download a file to a local computer over the Internet or convert a hardcopy image, such as a photograph, into digital by scanning the image with any of the common image scanners.. Then start an image manipulation program that supports watermarking, and load the image file from a file selection menu. Select the menu item to add a watermark, and save the watermarked image into a file.

To recover a watermarked image from a digital image on, say, a PC, just as in the case of embedding a watermark, convert the image to a suitable computer file format, scanning it in if needed. Then start either an image manipulation tool supporting watermarking, or start a stand-alone reader. Now open the image. In the case of a preferred embodiment image manipulation tool, the tool will automatically check for a watermark when the digital image file is opened, copied to the system clipboard or the image is scanned into the computer. If present, a copyright symbol is added to the title or status bar, or some other signal is provided to the user. By selecting a read watermark menu option, a user can discover the contents of the watermark. From the results dialog, a user can click on the Web Lookup button to retrieve complete contact information from MarcCentre, a network WWW on-line service. The application can be carried out with at least two kinds of watermarks. The first is a public watermark, which can be read by any suitably configured reader. This is the type of watermark supported by PictureMarc--Its purpose is communication. The second is a private watermark. Private watermarks are associated with a secret key used at the time the watermark is embedded in an image. Only the person with the secret key can find and read a private watermark. Private watermarks are used in proof of ownership applications. Both a public and a private watermark can coexist in an image.

If a user attempts to read a watermark and none is there, the reader portion will simply report that a watermark was not found in the image. If a user attempts to watermark an image already watermarked, the user will be informed there is already a watermark in the image, and

will be prevented from adding another watermark. In creating a composite image, the user is first told about the presence of a watermark in each watermarked original (assuming the use of a tool supporting watermarks). This addresses the first order problem of communicating copyright. The watermark will not prevent creating a composite work, in that a watermarked image, or a portion thereof, can be used as an element of a composite image. However, if the composite image contains multiple watermarked elements, the reader portion will tend to read the watermark from the most dominant piece. However, if the user suspects a component to be watermarked, or is simply interested in the source of an image element, the user can check for the watermark in that element by highlighting the area in question, and invoking the read watermark function. This provides a tool to inspect work to discover sources of individual elements. Copyright hierarchies in composited works and partnering with companies addressing rights management can all be handled in such a network application of a watermark.

Preferably, applications which include watermarking, such as Photoshop 4.0, support automatic detection ("autodetect"). This means that whenever an image is opened or scanned into the application, a quick check is done to determine if a watermark is present. If it is present, a visual indication, such as a copyright symbol, is added to the title or status bar.

The watermark placed by PictureMarc is a 128 bits long, with a data payload of 76 bits. The remaining bits are dedicated to header and control information, and error detection and correction. However, the bandwidth can be increased and consequently the data payload can also be increased. The detect process looks for the presence of a watermark, which completes in a couple of seconds, usually under 3 seconds regardless of image size, and the typical time to perform a full read watermark operation is under 15 seconds.

PictureMarc supports all the file formats of the host application, since it works with pixels in each of the major color schemes (CMYK, RGB, LAB, Grayscale). In the case of PictureMarc for Photoshop, all of the Photoshop file formats are supported.

Note that the watermark itself carries version and message type information, much like a communications message. PictureMarc has been designed to support upward compatibility. Any

Ex.1002 / Page 266 of 1311
APPLE INC.

suitably configured reader will be able to read previous versions of the watermark. If a reader finds a newer version watermark which it does not support, the user will be notified and told to download the most recent reader.

In the present application of PictureMarc, an image generally needs to be at least 256x256 pixels to carry a watermark. Often a watermark can be found in patches as small as 100 pixels square. Arbitrarily large images can be watermarked, constrained only by available memory and disk space. The watermark does not add data to the image file, since it only makes very small changes to the luminance of pixels. This can change the size of a compressed image by a few percent, but the size of uncompressed images, such as bitmaps (.BMPs), are unchanged. While a watermark survives compression, to watermark an already compressed image, it is preferable to either work from the original, or uncompress, watermark, and recompress the image. Future versions will likely support adding watermarks directly to the JPEG compressed file.

PictureMarc supports the automation features provided by Adobe Photoshop 4.0. This means that repetitive watermarking operations can be automated now. In other applications, support can be provided for unattended batch processing of images, aimed to support the needs of image distributors, especially those companies looking to watermark images as they are distributed with unique Transition IDs.

As to the use of a network on line service, an image creator or distributor subscribes to the MarcCentre, on-line locator service. Subscribing includes providing a set of contact information to be made available when one of a user's watermarked images is found. Contact details include name, address, phone number, specialty, intermediary (e.g., organization, stock agency, representative, etc.) representing the creator's work and so on. When subscribing, the user is given a unique Creator ID. The user provides this Creator ID to a copy of PictureMarc, one time when first using the software-based system. This Creator ID links all of the user's images to facilitate contact details. Information can be queried via the Internet, or via a fax-back service. Querying is preferably free.

A subscription to MarcCentre is priced to include unlimited access to contact information by anyone finding watermarked images. A Creator ID in the image leverages the power of a network accessible database, and lets the image creator communicate much more information than could fit in the image. In essence, we are creating a copyright communication system. Watermark detection in programs such as Photoshop is automatic. The Creator ID in the image, along with the image attributes, uniquely identify the image creator or distributor. The contact information corresponding to the Creator ID is always up to date, no matter how long ago a watermarked image was distributed. To make this copyright communication system work, we need unique ids which correspond to the appropriate image creator contact information.

If there is no network connection for PictureMarc, an equivalent approach can be used: contact information can be communicated via a fax-back service. This can be carried out by punching in the Creator ID on a touch-tone keypad with telephony, and the resulting contact details for that image creator will be sent by fax.

Contact details will be stored on a robust server being hosted by an Internet provider. The Web site is responsible for maintaining the locator service and ensuring that information is available. Individual image creators are responsible for maintaining their own contact details, keeping them current as information changes. This ensures that all of the images distributed point to up-to-date information about the creator and the creator's work.

Data in the Web site repository is stored in a way that when a user enter contact details, the user has control over which pieces of information are viewable by viewers. Thus, for example, the user can designate whether to exclude his or her address, phone number, and other specific information.

It is helpful for users to receive regular reports, generated by the Web page site software, about the number of people querying a specific user's information. These can be sent to the user via e-mail or fax, to help communicate the activity concerning watermarked images.

Before turning to a specific embodiment of the present invention, it is important to stress the general applicability of embedding a watermark, then reading it to obtain data, and then using the data to link to a database--preferably via a computer network. This is applicable to all media for which this technology is useful. This includes watermarking a photograph or a series of watermarks in a series of images (e.g., television, video), in sound or other analog signals (e.g., recordings, cellular telephone or other broadcasts), and other applications indicated herein. In any of these applications, while the watermark can contain self-identifying data, a centralized repository of watermark data facilitates broad, efficient usage.

Thus, the following specific description of a preferred embodiment of a central repository, accessible by a network, is intended to be illustrative of the same kind of an approach for other applications and media suitable for watermarking. And it is to be understood that while the following discussion is made in the context of image processing, the same general approach can be used for handling watermarks in acoustic or other applications. Also, while the Adobe Photoshop application is used to illustrate the generally applicable concepts, it should be appreciated that the present invention can be used in connection with other such products from other vendors, as well as in stand alone form.

Fig. 43 illustrates an application of the invention using such an application as Adobe Photoshop 4.0, equipped with a plug in version of the digital watermarking technology discussed herein and illustrated as "PictureMarc," running on a digital electrical computer. The computer running the Adobe application is adapted for communication over a network (such as the WWW) with another digital electrical computer, running the MarcCentre Web Page computer program.

At step 1 of Fig. 43 a creator of a digitized photographic image, the "user" of the Adobe Photoshop, commences by obtaining a Creator ID. Such data is created and stored in a central repository designated in Fig. 43 as MarcCentre Locator Service, which is connectable to PictureMarc via Internet or fax. At step 2 of Fig. 43, the user creates and embeds a watermark in the digital image. At step 3 of Fig. 43, that image can be stored and/or printed, possibly subsequent to distribution over a computer system such as the Internet or WWW. When the image bearing the watermark is used in digital form, at step 4 of Fig. 43, that image is examined

for a watermark by a computer running the PictureMarc computer program. Preferably, the examination is conducted automatically (rather than selectively) upon opening an image, e.g., upon locating the image on a computer screen clipboard. If a watermark is detected, at step 5 of Fig. 43, PictureMarc communicates with the MarcCentre Locator Service to look up creator information and present a WWW page to the user.

The means by which a user obtains a Creator ID in step 1 of Fig. 43 is elaborated in Figs. 44, 45, and 46, which represent screen images of MarcCentre, a Web Page locatable on the WWW. Introductory information is set forth in Fig. 44. Note the mention of a single mouse click. This feature refers to a hot button, such as a copyright symbol in a title bar of a screen, which permits someone using the Web Page to obtain information about the image that was pre-specified by the creator. (A hot button (pointer to information stored in a MarcCentre database, which can in turn have a pointer to another database, such as that accessed by means of the network or WWW.)

Figs. 45 and 46 show sequential screens of a "form" to be completed by computer entries. This information is stored in a MarcCentre database. Note too that the information entered to complete the form include a fee, and the fee account for the user is also stored in a MarcCentre database. After the form has been completed and stored and the fee has been secured, a Creator ID is issued by the MarcCentre service.

The means by which a user with a Creator ID embeds a registered watermark in step 2 of Fig. 43 is elaborated in Figs. 47, 48, and 49. The Photoshop software uses a filter to call up the "Digimarc" PictureMarc software and select options to embed a watermark. Bolded bars on the screen of Fig. 47 illustrate a selection being made for a digital image in the background. Other information on the screen is typical for a windows application, the Adobe Photoshop application, and or generally known information about digital images, e.g., 100% shows the scale of the image, "RGB" indicates the image format, etc.

Fig. 48 illustrates a "Dialog Box" used in associating information in connection with the embedding of the watermark. In a portion of the screen, the Copyright Information is

specified, including the Creator ID, the Type of Use (either Restricted or Royalty Free), Adult Content, and a Personalize option. A demonstration ID for use by those who do not have a registered Creator ID is "PictureMarc Demo." The Adult Content feature permits embedding in the watermark data, preferably content-indicative data. Here the data is used for identifying that the photograph contains adult content, and thus a subsequent use of the photograph can be restricted. However, an alternative is to include other data or other content-identifying data, such as cataloging information, in the watermark itself. Cataloging information permits subject matter searching in a collection of photographs. Another feature shown in Fig. 48 is a slidable scale for adjusting the intensity of the watermark. The more visible the watermark, the more durable the watermark is. Watermark durability (visibility) is also indicated by a number shown on the screen. Once the screen has been completed, if it is "OK", the user can click on this button and thereby embed the watermark. Otherwise, the user can select to cancel and thereby abort the process that would have led to the watermarking.

Selecting the Personalize option of Fig. 48 calls the screen illustrated in Fig. 49, which is used to obtain a personalized, registered Creator ID. After the Creator ID is entered, selection of the Register option launches a WWW browser directed by the URL address shown entered on the screen to the MarcCentre registration page.

In a preferred embodiment of step 4 of Fig. 43, each time an image is opened, scanned in, or copied to the clipboard by any application, the image is automatically tested for a watermark, as illustrated in Fig. 50. This "autodetect" approach generates a signal indicative of the result of the test, for example, one or more copyright symbols shown in the title page of the screen in Fig. 50. Automatic checking for a watermark takes less computer time than the alternative of selectively reading the watermark, which is illustrated in Figs. 51 and 52. However, the selective approach is less comprehensive in monitoring the use of watermarked images, which is disadvantageous if the application is configured to automatically report the detection back to the MarcCentre. In Fig. 50, the copyright symbol in the title area of the photograph is the previously mentioned hot button. The copyright symbol in the lower left of the screen can alternatively or also be a hot button.

Figs. 51 and 52 elaborate the alternative approach to step 4 of Fig. 43 in which an image is opened and examined for a watermark. Preferably, each time an image is opened, there is an automatic check for a watermark. In any case, Fig. 51 is similar to the screen of Fig. 47, except that a selection is shown to read for a watermark.

Fig. 52 shows a screen that is be produced in response to the reading a watermark. This screen provides the Creator ID of the owner and / or distributor of this image, as well as whether use of this image is restricted or royalty free. Note the Web Lookup button, which can be selected to launch a WWW browser directed by the URL address shown entered on the screen to the MarcCentre Web Page, and obtains the data in the corresponding MarcCentre database corresponding to the Creator ID.

Fig. 53 illustrates a screen produced from selecting the Web Lookup feature of Fig. 53. This screen provides the contact information for the image creator, including phone number, E-mail address and Web links. Note the selectable buttons at the bottom of the screen illustrated in Fig. 53, including buttons that take the user to promotional offers, allow the user to download additional software, subscribe to the MarcCentre service, update contact information, search for a creator or provide feedback on the MarcCentre service. Selecting the Creator Search option leads to the screen shown in Fig. 54 This self-explanatory screen enables a user to obtain contact information about a particular creator by entering the Creator ID.

In a variation on the above-described approach, extended information is embedded in the watermark and subsequently detected, as respectively illustrated in Figs.' 55 and 56. Such a feature would be inserted as an additional filter in Fig. 47. In Fig. 55., the dialog box permits inputting an Organization ID, an Item ID, and a Transaction ID. An Organization ID identifies an organization or other intermediary representing the creator and/or authorized for distributing the photograph and licensing rights thereto, such as a photo stock house. An Item ID identifies the photograph in a collection, such as a CD ROM. A Transaction ID identifies the specific transaction that is authorized (e.g., publication one time only in a newspaper), which permits detection of an authorized usage. The other feature on the screen, The Bump Size specification, pertains to the smallest unit of data, in the horizontal and vertical directions, that the watermark algorithms operate on. The extended information embedded in the watermark can be detected in

a manner analogous to that described with respect to Fig. 48, except that the additional information is shown as illustrated in Fig. 56.

An alternative to a "plug in" embodiment is a stand alone version, which is exemplified in the next sequence of figures. As discussed above, for any application that involves a watermark, preferably there is an autodetect capability, and Fig. 57 illustrates a screen from a stand alone image reader with a clip board triggered by merely putting an image on the clipboard. In the Clipboard Viewer window of Fig. 57, a watermark detected on the image can be used to signal the user by one of several methods that can be used individually or in a combination. In one method, a title bar icon or some other bar or icon alternates between "normal" and "watermark detected" states periodically (for example, at one second intervals). Another approach is to use other state changes, like having the title bar or a task bar alternate between highlighted and normal states periodically. Still another method is to have the detection initiate a special dialog box display, as illustrated in Fig. 58. The Read button in the special dialog box (triggered by a click of a mouse, like other buttons) causes reading and display of the watermark information in the image. The Close button ends the special dialog. (So as not to interfere with image processing, it can be convenient to show the special dialog box for a limited amount of time, and a small window therein can show the time remaining before the special dialog box disappears.) Buttons shown on the screen include one resembling a file folder to open a file; the next button is for copying a selected image from a first application to the clipboard; the next button is for pasting an image from the clipboard to a second application; and the last is for help menus.

Fig. 59 shows the image after pasting it from the clipboard, in the course of image processing. The Digimarc Reader - Image 1 window pops up on the screen to permit clicking the Web lookup button, which launches the user's web browser to result in a display of the creator's web page on the MarcCentre Web site, with the rest following as above.

Note again that the foregoing approach is representative, and it can be modified for another media in which a watermark is used. For example, instead of an image, a sound can be used.

POTENTIAL USE OF THE TECHNOLOGY IN THE PROTECTION AND CONTROL OF SOFTWARE PROGRAMS

The illicit use, copying, and reselling of software programs represents a huge loss of revenues to the software industry at large. The prior art methods for attempting to mitigate this problem are very broad and will not be discussed here. What will be discussed is how the principles of this technology might be brought to bear on this huge problem. It is entirely unclear whether the tools provided by this technology will have any economic advantage (all things considered) over the existing countermeasures both in place and contemplated.

The state of technology over the last decade or more has made it a general necessity to deliver a full and complete copy of a software program in order for that program to function on a user's computer. In effect, $X were invested in creating a software program where X is large, and the entire fruits of that development must be delivered in its entirety to a user in order for that user to gain value from the software product. Fortunately this is generally compiled code, but the point is that this is a shaky distribution situation looked at in the abstract. The most mundane (and harmless in the minds of most perpetrators) illicit copying and use of the program can be performed rather easily.

This disclosure offers, at first, an abstract approach which may or may not prove to be economical in the broadest sense (where the recovered revenue to cost ratio would exceed that of most competing methods, for example). The approach expands upon the methods and approaches already laid out in the section on plastic credit and debit cards. The abstract concept begins by positing a "large set of unique patterns," unique among themselves, unique to a given product, and unique to a given purchaser of that product. This set of patterns effectively contains thousands and even millions of absolutely unique "secret keys" to use the cryptology vernacular. Importantly and distinctly, these keys are non-deterministic, that is, they do not arise from singular sub-1000 or sub-2000 bit keys such as with the RSA key based systems. This large set of patterns is measured in kilobytes and Megabytes, and as mentioned, is non-deterministic in

nature. Furthermore, still at the most abstract level, these patterns are fully capable of being encrypted via standard techniques and analyzed within the encrypted domain, where the analysis is made on only a small portion of the large set of patterns, and that even in the worst case scenario where a would-be pirate is monitoring the step-by-step microcode instructions of a microprocessor, this gathered information would provide no useful information to the would-be pirate. This latter point is an important one when it comes to "implementation security" as opposed to "innate security" as will be briefly discussed below.

So what could be the differential properties of this type of key based system as opposed to, for example, the RSA cryptology methods which are already well respected, relatively simple, etc. etc? As mentioned earlier, this discussion is not going to attempt a commercial side-by-side analysis. Instead, we'll just focus on the differing properties. The main distinguishing features fall out in the implementation realm (the implementation security). One example is that in single low-bit-number private key systems, the mere local use and re-use of a single private key is an inherently weak link in an encrypted transmission system. ["Encrypted transmission systems" are discussed here in the sense that securing the paid-for use of software programs will in this discussion require de facto encrypted communication between a user of the software and the "bank" which allows the user to use the program; it is encryption in the service of electronic financial transactions looked at in another light.] Would-be hackers wishing to defeat so-called secure systems never attack the fundamental hard-wired security (the innate security) of the pristine usage of the methods, they attack the implementation of those methods, centered around human nature and human oversights. It is here, still in the abstract, that the creation of a much larger key base, which is itself non-deterministic in nature, and which is more geared toward effectively throw-away keys, begins to "idiot proof" the more historically vulnerable implementation of a given secure system. The huge set of keys is not even comprehensible to the average holder of those keys, and their use of those keys (i.e., the "implementation" of those keys) can randomly select keys, easily throw them out after a time, and can use them in a way that no "eavesdropper" will gain any useful information in the eavesdropping, especially when well within a millionth of the amount of time that an eavesdropper could "decipher" a key, its usefulness in the system would be long past.

Turning the abstract to the semi-concrete, one possible new approach to securely delivering a software product to ONLY the bonafide purchasers of that product is the following. In a mass economic sense, this new method is entirely founded upon a modest rate realtime digital connectivity (often, but not necessarily standard encrypted) between a user's computer network and the selling company's network. At first glance this smells like trouble to any good marketing person, and indeed, this may throw the baby out with the bathwater if by trying to recover lost revenues, you lose more legitimate revenue along the way (all part of the bottom line analysis). This new method dictates that a company selling a piece of software supplies to anyone who is willing to take it about 99.8% of its functional software for local storage on a user's network (for speed and minimizing transmission needs). This "free core program" is entirely unfunctional and designed so that even the craftiest hackers can't make use of it or "decompile it" in some sense. Legitimate activation and use of this program is performed purely on a instruction-cycle-count basis and purely in a simple very low overhead communications basis between the user's network and the company's network. A customer who wishes to use the product sends payment to the company via any of the dozens of good ways to do so. The customer is sent, via common shipment methods, or via commonly secured encrypted data channels, their "huge set of unique secret keys." If we were to look at this large set as if it were an image, it would look just like the snowy images discussed over and over again in other parts of this disclosure. (Here, the "signature" is the image, rather than being imperceptibly placed onto another image). The special nature of this large set is that it is what we might call "ridiculously unique" and contains a large number of secret keys. (The "ridiculous" comes from the simple math on the number of combinations that are possible with, say 1 Megabyte of random bit values, equaling exactly the number that "all ones" would give, thus 1 Megabyte being approximately 10 raised to the ~2,400,000 power, plenty of room for many people having many throwaway secret keys). It is important to re-emphasize that the purchased entity is literally: productive use of the tool. The marketing of this would need to be very liberal in its allotment of this productivity, since per-use payment schemes notoriously turn off users and can lower overall revenues significantly.

This large set of secret keys is itself encrypted using standard encryption techniques. The basis for relatively higher "implementation security" can now begin to manifest itself. Assume that the user now wishes to use the software product. They fire up the free core, and the free

core program finds that the user has installed their large set of unique encrypted keys. The core program calls the company network and does the usual handshaking. The company network, knowing the large set of keys belonging to that bonafide user, sends out a query on some simple set of patterns, almost exactly the same way as described in the section on the debit and credit cards. The query is such a small set of the whole, that the inner workings of the core program do not even need to decrypt the whole set of keys, only certain parts of the keys, thus no decrypted version of the keys ever exist, even within the machine cycles on the local computer itself. As can be seen, this does not require the "signatures within a picture" methods of the main disclosure, instead, the many unique keys ARE the picture. The core program interrogates the keys by performing certain dot products, then sends the dot products back to the company's network for verification. See Fig. 25 and the accompanying discussion for typical details on a verification transaction. Generally encrypted verification is sent, and the core program now "enables" itself to perform a certain amount of instructions, for example, allowing 100,000 characters being typed into a word processing program (before another unique key needs to be transmitted to enable another 100,000). In this example, a purchaser may have bought the number of instructions which are typically used within a one year period by a single user of the word processor program. The purchaser of this product now has no incentive to "copy" the program and give it to their friends and relatives.

All of the above is well and fine except for two simple problems. The first problem can be called "the cloning problem" and the second "the big brother problem." The solutions to these two problems are intimately linked. The latter problem will ultimately become a purely social problem, with certain technical solutions as mere tools not ends.

The cloning problem is the following. It generally applies to a more sophisticated pirate of software rather than the currently common "friend gives their distribution CD to a friend" kind of piracy. Crafty-hacker "A" knows that if she performs a system-state clone of the "enabled" program in its entirety and installs this clone on another machine, then this second machine effectively doubles the value received for the same money. Keeping this clone in digital storage, hacker "A" only needs to recall it and reinstall the clone after the first period is run out, thus indefinitely using the program for a single payment, or she can give the clone to their hacker friend "B" for a six-pack of beer. One good solution to this problem requires, again, a rather well

developed and low cost real time digital connectivity between user site and company enabling network. This ubiquitous connectivity generally does not exist today but is fast growing through the Internet and the basic growth in digital bandwidth. Part and parcel of the "enabling" is a negligible communications cost random auditing function wherein the functioning program routinely and irregularly performs handshakes and verifications with the company network. It does so, on average, during a cycle which includes a rather small amount of productivity cycles of the program. The resulting average productivity cycle is in general much less than the raw total cost of the cloning process of the overall enabled program. Thus, even if an enabled program is cloned, the usefulness of that instantaneous clone is highly limited, and it would be much more cost effective to pay the asking price of the selling company than to repeat the cloning process on such short time periods. Hackers could break this system for fun, but certainly not for profit. The flip side to this arrangement is that if a program "calls up" the company's network for a random audit, the allotted productivity count for that user on that program is accounted for, and that in cases where bonafide payment has not been received, the company network simply withholds its verification and the program no longer functions. We're back to where users have no incentive to "give this away" to friends unless it is an explicit gift (which probably is quite appropriate if they have indeed paid for it: "do anything you like with it, you paid for it").

The second problem of "big brother" and the intuitively mysterious "enabling" communications between a user's network and a company's network would as mentioned be a social and perceptual problem that should have all manner of potential real and imagined solutions. Even with the best and objectively unbeatable anti-big-brother solutions, there will still be a hard-core conspiracy theory crowd claiming it just ain't so. With this in mind, one potential solution is to set up a single program registry which is largely a public or non-profit institution to handling and coordinating the realtime verification networks. Such an entity would then have company clients as well as user clients. An organization such as the Software Publishers Association, for example, may choose to lead such an effort.

Concluding this section, it should be re-emphasized that the methods here outlined require a highly connected distributed system, in other words, a more ubiquitous and inexpensive Internet than exists in mid 1995. Simple trend extrapolation would argue that this is not too far off from

1995. The growth rate in raw digital communications bandwidth also argues that the above system might be more practical, sooner, than it might first appear. (The prospect of interactive TV brings with it the promise of a fast network linking millions of sites around the world.)

USE OF CURRENT CRYPTOLOGY METHODS IN CONJUNCTION WITH THIS TECHNOLOGY

It should be briefly noted that certain implementations of the principles of this technology probably can make good use of current cryptographic technologies. One case in point might be a system whereby graphic artists and digital photographers perform realtime registration of their photographs with the copyright office. It might be advantageous to send the master code signals, or some representative portion thereof, directly to a third party registry. In this case, a photographer would want to know that their codes were being transmitted securely and not stolen along the way. In this case, certain common cryptographic transmission might be employed. Also, photographers or musicians, or any users of this technology, may want to have reliable time stamping services which are becoming more common. Such a service could be advantageously used in conjunction with the principles of this technology.

DETAILS ON THE LEGITIMATE AND ILLEGITIMATE DETECTION AND REMOVAL OF INVISIBLE SIGNATURES

In general, if a given entity can recognize the signatures hidden within a given set of empirical data, that same entity can take steps to remove those signatures. In practice, the degree of difficulty between the former condition and the latter condition can be made quite large, fortunately. On one extreme, one could posit a software program which is generally very difficult to "decompile" and which does recognition functions on empirical data. This same bit of software could not generally be used to "strip" the signatures (without going to extreme lengths).

On the other hand, if a hacker goes to the trouble of discovering and understanding the "public codes" used within some system of data interchange, and that hacker knows how to recognize the signatures, it would not be a large step for that hacker to read in a given set of signed data and create a data set with the signatures effectively removed. In this latter example, interestingly enough, there would often be telltale statistics that signatures had been removed, statistics which will not be discussed here.

These and other such attempts to remove the signatures we can refer to as illicit attempts. Current and past evolution of the copyright laws have generally targeted such activity as coming under criminal activity and have usually placed such language, along with penalties and enforcement language, into the standing laws. Presumably any and all practitioners of this signature technology will go to lengths to make sure that the same kind of a) creation, b) distribution, and c) use of these kinds of illicit removal of copyright protection mechanisms are criminal offenses subject to enforcement and penalty. On the other hand, it is an object of this technology to point out that through the recognition steps outlined in this disclosure, software programs can be made such that the recognition of signatures can simply lead to their removal by inverting the known signatures by the amount equal to their found signal energy in the recognition process (i.e., remove the size of the given code signal by exact amount found). By pointing this out in this disclosure, it is clear that such software or hardware which performs this signature removal operation will not only (presumably) be criminal, but it will also be liable to infringement to the extent that it is not properly licensed by the holders of the (presumably) patented technology.

The case of legitimate and normal recognition of the signatures is straightforward. In one example, the public signatures could deliberately be made marginally visible (i.e. their intensity would be deliberately high), and in this way a form of sending out "proof comps" can be accomplished. "Comps" and "proofs" have been used in the photographic industry for quite some time, where a degraded image is purposely sent out to prospective customers so that they might evaluate it but not be able to use it in a commercially meaningful way. In the case of this technology, increasing the intensity of the public codes can serve as a way to "degrade" the commercial value intentionally, then through hardware or software activated by paying a purchase

price for the material, the public signatures can be removed (and possibly replaced by a new invisible tracking code or signature, public and/or private.

MONITORING STATIONS AND MONITORING SERVICES

Ubiquitous and cost effective recognition of signatures is a central issue to the broadest proliferation of the principles of this technology. Several sections of this disclosure deal with this topic in various ways. This section focuses on the idea that entities such as monitoring nodes, monitoring stations, and monitoring agencies can be created as part of a systematic enforcement of the principles of the technology. In order for such entities to operate, they require knowledge of the master codes, and they may require access to empirical data in its raw (unencrypted and untransformed) form. (Having access to original unsigned empirical data helps in finer analyses but is not necessary.)

Three basic forms of monitoring stations fall out directly from the admittedly arbitrarily defined classes of master codes: a private monitoring station, a semi-public, and a public. The distinctions are simply based on the knowledge of the master codes. An example of the fully private monitoring station might be a large photographic stock house which decides to place certain basic patterns into its distributed material which it knows that a truly crafty pirate could decipher and remove, but it thinks this likelihood is ridiculously small on an economic scale. This stock house hires a part-time person to come in and randomly check high value ads and other photography in the public domain to search for these relatively easy to find base patterns, as well as checking photographs that stock house staff members have "spotted" and think it might be infringement material. The part time person cranks through a large stack of these potential infringement cases in a few hours, and where the base patterns are found, now a more thorough analysis takes place to locate the original image and go through the full process of unique identification as outlined in this disclosure. Two core economic values accrue to the stock house in doing this, values which by definition will outweigh the costs of the monitoring service and the cost of the signing process itself. The first value is in letting their customers and the world know

that they are signing their material and that the monitoring service is in place, backed up by whatever statistics on the ability to catch infringers. This is the deterrent value, which probably will be the largest value eventually. A general pre-requisite to this first value is the actual recovered royalties derived from the monitoring effort and its building of a track record for being formidable (enhancing the first value).

The semi-public monitoring stations and the public monitoring stations largely follow the same pattern, although in these systems it is possible to actually set up third party services which are given knowledge of the master codes by clients, and the services merely fish through thousands and millions of "creative property" hunting for the codes and reporting the results to the clients. ASCAP and BMI have "lower tech" approaches to this basic service.

A large coordinated monitoring service using the principles of this technology would classify its creative property supplier clients into two basic categories, those that provide master codes themselves and wish the codes to remain secure and unpublished, and those that use generally public domain master codes (and hybrids of the two, of course). The monitoring service would perform daily samplings (checks) of publicly available imagery, video, audio, etc., doing high level pattern checks with a bank of supercomputers. Magazine ads and images would be scanned in for analysis, video grabbed off of commercial channels would be digitized, audio would be sampled, public Internet sites randomly downloaded, etc. These basic data streams would then be fed into an ever-churning monitoring program which randomly looks for pattern matches between its large bank of public and private codes, and the data material it is checking. A small sub-set, which itself will probably be a large set, will be flagged as potential match candidates, and these will be fed into a more refined checking system which begins to attempt to identify which exact signatures may be present and to perform a more fine analysis on the given flagged material. Presumably a small set would then fall out as flagged match material, owners of that material would be positively identified and a monitoring report would be sent to the client so that they can verify that it was a legitimate sale of their material. The same two values of the private monitoring service outlined above apply in this case as well. The monitoring service could also serve as a formal bully in cases of a found and proven infringement, sending out letters to infringing parties witnessing the found infringement and seeking inflated royalties so that the infringing party might avoid the more costly alternative of going to court.

METHOD FOR EMBEDDING SUBLIMINAL REGISTRATION PATTERNS INTO IMAGES AND OTHER SIGNALS

The very notion of reading embedded signatures involves the concept of registration. The underlying master noise signal must be known, and its relative position needs to be ascertained (registered) in order to initiate the reading process itself (e.g. the reading of the 1's and 0's of the N-bit identification word). When one has access to the original or a thumbnail of the unsigned signal, this registration process is quite straightforward. When one doesn't have access to this signal, which is often the case in universal code applications of this technology, then different methods must be employed to accomplish this registration step. The example of pre-marked photographic film and paper, where by definition there will never be an "unsigned" original, is a perfect case in point of the latter.

Many earlier sections have variously discussed this issue and presented certain solutions. Notably, the section on "simple" universal codes discusses one embodiment of a solution where a given master code signal is known a priori, but its precise location (and indeed, it existence or non-existence) is not known. That particular section went on to give a specific example of how a very low level designed signal can be embedded within a much larger signal, wherein this designed signal is standardized so that detection equipment or reading processes can search for this standardized signal even though its exact location is unknown. The brief section on 2D universal codes went on to point out that this basic concept could be extended into 2 dimensions, or, effectively, into imagery and motion pictures. Also, the section on symmetric patterns and noise patterns outlined yet another approach to the two dimensional case, wherein the nuances associated with two dimensional scale and rotation were more explicitly addressed. Therein, the idea was not merely to determine the proper orientation and scale of underlying noise patterns, but to have information transmitted as well, e.g., the N rings for the N-bit identification word.

This section now attempts to isolate the sub-problem of registering embedded patterns for registration's sake. Once embedded patterns are registered, we can then look again at how this registration can serve broader needs. This section presents yet another technique for embedding patterns, a technique which can be referred to as "subliminal digital graticules". "Graticules" - other words such as fiducials or reticles or hash marks could just as well be used -conveys the

idea of calibration marks used for the purposes of locating and/or measuring something. In this case, they are employed as low level patterns which serve as a kind of gridding function. That gridding function itself can be a carrier of 1 bit of information, as in the universal 1 second of noise (its absence or presence, copy me, don't copy me), or it can simply find the orientation and scale of other information, such as embedded signatures, or it can simply orient an image or audio object itself.

Figs. 29 and 30 visually summarize two related methods which illustrate applicant's subliminal digital graticules. As will be discussed, the method of Fig. 29 may have slight practical advantages over the method outlined in Fig. 30, but both methods effectively decompose the problem of finding the orientation of an image into a series of steps which converge on a solution. The problem as a whole can be simply stated as the following: given an arbitrary image wherein a subliminal digital graticule may have been stamped, then find the scale, rotation, and origin (offset) of the subliminal digital graticule.

The beginning point for subliminal graticules is in defining what they are. Simply put, they are visual patterns which are directly added into other images, or as the case may be, exposed onto photographic film or paper. The classic double exposure is not a bad analogy, though in digital imaging this specific concept becomes rather stretched. These patterns will generally be at a very low brightness level or exposure level, such that when they are combined with "normal" images and exposures, they will effectively be invisible (subliminal) and just as the case with embedded signatures, they will by definition not interfere with the broad value of the images to which they are added.

Figs. 29 and 30 define two classes of subliminal graticules, each as represented in the spatial frequency domain, also known as the UV plane, 1000. Common two dimensional fourier transform algorithms can transform any given image into its UV plane conjugate. To be precise, the depictions in Figs. 29 and 30 are the magnitudes of the spatial frequencies, whereas it is difficult to depict the phase and magnitude which exists at every point.

Fig. 29 shows the example of six spots in each quadrant along the 45 degree lines, 1002. These are exaggerated in this figure, in that these spots would be difficult to discern by visual inspection of the UV plane image. A rough depiction of a "typical" power spectrum of an arbitrary image as also shown, 1004. This power spectrum is generally as unique as images are unique. The subliminal graticules are essentially these spots. In this example, there are six spatial frequencies combined along each of the two 45 degree axes. The magnitudes of the six frequencies can be the same or different (we'll touch upon this refinement later). Generally speaking, the phases of each are different from the others, including the phases of one 45 degree axis relative to the other. Fig. 31 depicts this graphically. The phases in this example are simply randomly placed between PI and -PI, 1008 and 1010. Only two axes are represented in Fig. 31 - as opposed to the four separate quadrants, since the phase of the mirrored quadrants are simply PI/2 out of phase with their mirrored counterparts. If we turned up the intensity on this subliminal graticule, and we transformed the result into the image domain, then we would see a weave-like cross-hatching pattern as related in the caption of Fig. 29. As stated, this weave-like pattern would be at a very low intensity when added to a given image. The exact frequencies and phases of the spectral components utilized would be stored and standardized. These will become the "spectral signatures" that registration equipment and reading processes will seek to measure.

Briefly, Fig. 30 has a variation on this same general theme. Fig. 30 lays out a different class of graticules in that the spectral signature is a simple series of concentric rings rather than spots along the 45 degree axes. Fig. 32 then depicts a quasi-random phase profile as a function along a half-circle (the other half of the circle then being PI/2 out of phase with the first half). These are simple examples and there are a wide variety of variations possible in designing the phase profiles and the radii of the concentric rings. The transform of this type of subliminal graticule is less of a "pattern" as with the weave-like graticule of Fig. 29, where it has more of a random appearance like a snowy image.

The idea behind both types of graticules is the following: embed a unique pattern into an image which virtually always will be quite distinct from the imagery into which it will be added, but which has certain properties which facilitate fast location of the pattern, as well as accuracy properties such that when the pattern is generally located, its precise location and orientation can be found to some high level of precision. A corollary to the above is to design the pattern such

that the pattern on average minimally interferes with the typical imagery into which it will be added, and has maximum energy relative to the visibility of the pattern.

Moving on to the gross summary of how the whole process works, the graticule type of Fig. 29 facilitates an image processing search which begins by first locating the rotation axes of the subliminal graticule, then locating the scale of the graticule, then determining the origin or offset. The last step here identifies which axes is which of the two 45 degree axes by determining phase. Thus even if the image is largely upside down, an accurate determination can be made. The first step and the second step can both be accomplished using only the power spectrum data, as opposed to the phase and magnitude. The phase and magnitude signals can then be used to "fine tune" the search for the correct rotation angle and scale. The graticule of Fig. 30 switches the first two steps above, where the scale is found first, then the rotation, followed by precise determination of the origin. Those skilled in the art will recognize that determining these outstanding parameters, along two axes, are sufficient to fully register an image. The "engineering optimization challenge" is to maximize the uniqueness and brightness of the patterns relative to their visibility, while minimizing the computational overhead in reaching some specified level of accuracy and precision in registration. In the case of exposing photographic film and paper, clearly an additional engineering challenge is the outlining of economic steps to get the patterns exposed onto the film and paper in the first place, a challenge which has been addressed in previous sections.

The problem and solution as above defined is what was meant by registration for registration's sake. It should be noted that there was no mention made of making some kind of value judgement on whether or not a graticule is indeed being found or not. Clearly, the above steps could be applied to images which do not in fact have graticules inside them, the measurements then simply chasing noise. Sympathy needs to be extended to the engineer who is assigned the task of setting "detection thresholds" for these types of patterns, or any others, amidst the incredibly broad range of imagery and environmental conditions in which the patterns must be sought and verified. [Ironically, this is where the pure universal one second of noise stood in a previous section, and that was why it was appropriate to go beyond merely detecting or not detecting this singular signal, i.e. adding additional information planes]. Herein is where some real beauty shows up: in the combination of the subliminal graticules with the now-registered

embedded signatures described in other parts of this disclosure. Specifically, once a "candidate registration" is found - paying due homage to the idea that one may be chasing noise - then the next logical step is to perform a reading process for, e.g., a 64 bit universal code signature. As further example, we can imagine that 44 bits of the 64 bit identification word are assigned as an index of registered users -- serial numbers if you will. The remaining 20 bits are reserved as a hash code - as is well known in encryption arts - on the 44 bit identification code thus found. Thus, in one swoop, the 20 bits serve as the "yes, I have a registered image" or "no, I don't" answer. More importantly, perhaps, this allows for a system which can allow for maximum flexibility in precisely defining the levels of "false positives" in any given automated identification system. Threshold based detection will always be at the mercy of a plethora of conditions and situations, ultimately resting on arbitrary decisions. Give me N coin flips any day.

Back on point, these graticule patterns must first be added to an image, or exposed onto a piece of film. An exemplary program reads in an arbitrarily sized digital image and adds a specified graticule to the digital image to produce an output image. In the case of film, the graticule pattern would be physically exposed onto the film either before, during, or after exposure of the primary image. All of these methods have wide variations in how they might be accomplished.

The searching and registering of subliminal graticules is the more interesting and involved process. This section will first describe the elements of this process, culminating in the generalized flow chart of Fig. 37.

Fig. 33 depicts the first major "search" step in the registration process for graticules of the type in Fig. 29. A suspect image (or a scan of a suspect photograph) is first transformed in its fourier representation using well known 2D FFT routines. The input image may look like the one in Fig. 36, upper left image. Fig. 33 conceptually represents the case where the image and hence the graticules have not been rotated, though the following process fully copes with rotation issues. After the suspect image has been transformed, the power spectrum of the transform is then calculated, being simply the square root of the addition of the two squared moduli. it is also a good idea to perform a mild low pass filter operation, such as a 3x3 blur filter, on the resulting power spectrum data. so that later search steps don't need incredibly fine spaced steps. Then the

candidate rotation angles from 0 through 90 degrees (or 0 to PI/2 in radian) are stepped through. Along any given angle, two resultant vectors are calculated, the first is the simple addition of power spectrum values at a given radius along the four lines emanating from the origin in each quadrant. The second vector is the moving average of the first vector. Then, a normalized power profile is calculated as depicted in both 1022 and 1024, the difference being that one plot is along an angle which does not align with the subliminal graticules, and the other plot does align. The normalization stipulates that the first vector is the numerator and the second vector is the denominator in the resultant vector. As can be seen in Fig. 33, 1022 and 1024, a series of peaks (which should be "six" instead of "five" as is drawn) develops when the angle aligns along its proper direction. Detection of these peaks can be effected by setting some threshold on the normalized values, and integrating their total along the whole radial line. A plot, 1026, from 0 to 90 degrees is depicted in the bottom of Fig. 33, showing that the angle 45 degrees contains the most energy. In practice, this signal is often much lower than that depicted in this bottom figure, and instead of picking the highest value as the "found rotation angle," one can simply find the top few candidate angles and submit these candidates to the next stages in the process of determining the registration. It can be appreciated by those practiced in the art that the foregoing was simply a known signal detection scheme, and that there are dozens of such schemes that can ultimately be created or borrowed. The simple requirement of the first stage process is to whittle down the candidate rotation angles to just a few, wherein more refined searches can then take over.

Fig. 34 essentially outlines the same type of gross searching in the power spectral domain. Here instead we first search for the gross scale of the concentric rings, stepping from a small scale through a large scale, rather than the rotation angle. The graph depicted in 1032 is the same normalized vector as in 1022 and 1024, but now the vector values are plotted as a function of angle around a semi-circle. The moving average denominator still needs to be calculated in the radial direction, rather than the tangential direction. As can be seen, a similar "peaking" in the normalized signal occurs when the scanned circle coincides with a graticule circle, giving rise to the plot 1040. The scale can then be found on the bottom plot by matching the known characteristics of the concentric rings (i.e. their radii) with the profile in 1040.

Fig. 35 depicts the second primary step in registering subliminal graticules of the type in Fig. 29. Once we have found a few rotation candidates from the methods of Fig. 33, we then

take the plots of the candidate angles of the type of 1022 and 1024 and perform what the inventor refers to as a "scaled kernel" matched filtering operation on those vectors. The scaled kernel refers to the fact that the kernel in this case is a known non-harmonic relationship of frequencies, represented as the lines with x's at the top in 1042 and 1044, and that the scale of these frequencies is swept through some pre-determined range, such as 25% to 400% of some expected scale at 100%. The matched filter operation simply adds the resultant multiplied values of the scaled frequencies and their plot counterparts. Those practiced in the art will recognize the similarity of this operation with the very well known matched filter operation. The resulting plot of the matched filter operation will look something like 1046 at the bottom of Fig. 35. Each candidate angle from the first step will generate its own such plot, and at this point the highest value of all of the plots will become our candidate scale, and the angle corresponding to the highest value will become our primary candidate rotation angle. Likewise for graticules of the type in Fig. 30, a similar "scaled-kernel" matched filtering operation is performed on the plot 1040 of Fig. 34. This generally provides for a single candidate scale factor. Then, using the stored phase plots 1012, 1014 and 1016 of Fig. 32, a more traditional matched filtering operation is applied between these stored plots (as kernels), and the measured phase profiles along the half-circles at the previously found scale.

The last step in registering graticules of the type in Fig. 29 is to perform a garden variety matched filter between the known (either spectral or spatial) profile of the graticule with the suspect image. Since both the rotation, scale and orientation are now known from previous steps, this matched filtering operation is straightforward. If the accuracies and precision of preceding steps have not exceeded design specifications in the process, then a simple micro-search can be performed in the small neighborhood about the two parameters scale and rotation, a matched filter operation performed, and the highest value found will determine a "fine tuned" scale and rotation. In this way, the scale and rotation can be found to within the degree set by the noise and cross-talk of the suspect image itself. Likewise, once the scale and rotation of the graticules of the type in Fig. 30 are found, then a straightforward matched filter operation can complete the registration process, and similar "fine tuning" can be applied.

Moving on to a variant of the use of the graticules of the type in Fig. 29, Fig. 36 presents the possibility for finding the subliminal graticules without the need for performing a

computationally expensive 2 dimensional FFT (fast fourier transform). In situations where computational overhead is a major issue, then the search problem can be reduced to a series of one-dimensional steps. Fig. 36 broadly depicts how to do this. The figure at the top left is an arbitrary image in which the graticules of the type of Fig. 29 have been embedded. Starting at angle 0, and finishing with an angle just below 180 degrees, and stepping by, for example 5 degrees, the grey values along the depicted columns can be simply added to create a resulting column-integrated scan, 1058. The figure in the top right, 1052, depicts one of the many angles at which this will be performed. This column-integrated scan then is transformed into its fourier representation using the less computationally expensive 1 dimensional FFT. This is then turned into a magnitude or "power" plot (the two are different), and a similar normalized vector version created just like 1022 and 1024 in Fig. 33. The difference now is that as the angle approaches the proper angles of the graticules, slowly the tell-tale peaks begin to appear in the 1024-like plots, but they generally show up at higher frequencies than expected for a given scale, since we are generally slightly off on our rotation. It remains to find the angle which maximizes the peak signals, thus zooming in on the proper rotation. Once the proper rotation is found, then the scaled kernel matched filter process can be applied, followed by traditional matched filtering, all as previously described. Again, the sole idea of the "short-cut" of Fig. 36 is to greatly reduce the computational overhead in using the graticules of the type in Fig. 29. The inventor has not reduced this method of Fig. 36 to practice and currently has no data on precisely how much computational savings, if any, will be realized. These efforts will be part of application specific development of the method.

Fig. 37 simply summarizes, in order of major process steps, the methods revolving around the graticules of the type in Fig. 29.

In another variant embodiment, the graticule energy is not concentrated around the 45 degree angles in the spatial frequency domain. (Some compression algorithms, such as JPEG, tend to particularly attenuate image energy at this orientation.) Instead, the energy is more widely spatially spread. Fig. 29A shows one such distribution. The frequencies near the axes, and near the origin are generally avoided, since this is where the image energy is most likely concentrated.

Detection of this energy in a suspect image again relies on techniques like that reviewed above. However, instead of first identifying the axes, then the rotation, and then the scale, a more global pattern matching procedure is performed in which all are determined in a brute force effort. Those skilled in the art will recognize that the Fourier-Mellin transform is well suited for use in such pattern matching problems.

The foregoing principles find application, for example, in photo-duplication kiosks. Such devices typically include a lens for imaging a customer-provided original (e.g. a photographic print or film) onto an opto-electronic detector, and a print-writing device for exposing and developing an emulsion substrate (again photographic paper or film) in accordance with the image data gathered by the detector. The details of such devices are well known to those skilled in the art, and are not belabored here.

In such systems, a memory stores data from the detector, and a processor (e.g. a Pentium microprocessor with associated support components) can be used to process the memory data to detect the presence of copyright data steganographically encoded therein. If such data is detected, the print-writing is interrupted.

To avoid defeat of the system by manual rotation of the original image off-axis, the processor desirably implements the above-described technique to effect automatic registration of the original, notwithstanding scale, rotation, and origin offset factors. If desired, a digital signal processing board can be employed to offload certain of the FFT processing from the main (e.g. Pentium) processor. After a rotated/scaled image is registered, detection of any steganographically encoded copyright notice is straightforward and assures the machine will not be used in violation of a photographer's copyright.

While the techniques disclosed above have make use of applicant's preferred steganographic encoding methods, the principles thereof are more widely applicable and can be used in many instances in which automatic registration of an image is to be effected.

USE OF EMBEDDED SIGNATURES IN VIDEO, WHEREIN A VIDEO DATA STREAM EFFECTIVELY SERVES AS A HIGH-SPEED ONE WAY MODEM

Through use of the universal coding system outlined in earlier sections, and through use of master snowy frames which change in a simple fashion frame to frame, a simple receiver can be designed such that it has pre-knowledge of the changes in the master snowy frames and can therefore read a changing N-bit message word frame by frame (or I-frame by I-frame as the case may be in MPEG video). In this way, a motion picture sequence can be used as a high speed one-way information channel, much like a one-way modem. Consider, for example, a frame of video data with N scan lines which is steganographically encoded to effect the transmission of an N-bit message. If there are 484 scan lines in a frame (N), and frames change 30 times a second, an information channel with a capacity comparable to a 14.4 kilobaud modem is achieved.

In actual practice, a data rate substantially in excess of N bits per frame can usually be achieved, yielding transmission rates nearer that of ISDN circuits.

FRAUD PREVENTION IN WIRELESS COMMUNICATIONS

In the cellular telephone industry, hundreds of millions of dollars of revenue is lost each year through theft of services. While some services are lost due to physical theft of cellular telephones, the more pernicious threat is posed by cellular telephone hackers.

Cellular telephone hackers employ various electronic devices to mimic the identification signals produced by an authorized cellular telephone. (These signals are sometimes called authorization signals, verification numbers, signature data, etc.) Often, the hacker learns of these signals by eavesdropping on authorized cellular telephone subscribers and recording the data exchanged with the cell cite. By artful use of this data, the hacker can impersonate an authorized subscriber and dupe the carrier into completing pirate calls.

In the prior art, identification signals are segregated from the voice signals. Most commonly, they are temporally separated, e.g. transmitted in a burst at the time of call origination. Voice data passes through the channel only after a verification operation has taken place on this identification data. (Identification data is also commonly included in data packets sent during the transmission.) Another approach is to spectrally separate the identification, e.g. in a spectral subband outside that allocated to the voice data.

Other fraud-deterrent schemes have also been employed. One class of techniques monitors characteristics of a cellular telephone's RF signal to identify the originating phone. Another class of techniques uses handshaking protocols, wherein some of the data returned by the cellular telephone is based on an algorithm (e.g. hashing) applied to random data sent thereto.

Combinations of the foregoing approaches are also sometimes employed.

U.S. Patents 5,465,387, 5,454,027, 5,420,910, 5,448,760, 5,335,278, 5,345,595, 5,144,649, 5,204,902, 5,153,919 and 5,388,212 detail various cellular telephone systems, and fraud deterrence techniques used therein. The disclosures of these patents are incorporated by reference.

As the sophistication of fraud deterrence systems increases, so does the sophistication of cellular telephone hackers. Ultimately, hackers have the upper hand since they recognize that all prior art systems are vulnerable to the same weakness: the identification is based on some attribute of the cellular telephone transmission outside the voice data. Since this attribute is segregated from the voice data, such systems will always be susceptible to pirates who electronically "patch" their voice into a composite electronic signal having the attribute(s) necessary to defeat the fraud deterrence system.

To overcome this failing, preferred embodiments of this aspect of the present technology steganographically encode the voice signal with identification data, resulting in "in-band" signalling (in-band both temporally and spectrally). This approach allows the carrier to monitor the user's voice signal and decode the identification data therefrom.

In one such form of the technology, some or all of the identification data used in the prior art (e.g. data transmitted at call origination) is repeatedly steganographically encoded in the user's voice signal as well. The carrier can thus periodically or aperiodically check the identification data accompanying the voice data with that sent at call origination to ensure they match. If they do not, the call is identified as being hacked and steps for remediation can be instigated such as interrupting the call.

In another form of the technology, a randomly selected one of several possible messages is repeatedly steganographically encoded on the subscriber's voice. An index sent to the cellular carrier at call set-up identifies which message to expect. If the message steganographically decoded by the cellular carrier from the subscriber's voice does not match that expected, the call is identified as fraudulent.

In a preferred form of this aspect of the technology, the steganographic encoding relies on a pseudo random data signal to transform the message or identification data into a low level noise-like signal superimposed on the subscriber's digitized voice signal. This pseudo random data signal is known, or knowable, to both the subscriber's telephone (for encoding) and to the cellular carrier (for decoding). Many such embodiments rely on a deterministic pseudo random number generator seeded with a datum known to both the telephone and the carrier. In simple embodiments this seed can remain constant from one call to the next (e.g. a telephone ID number). In more complex embodiments, a pseudo-one-time pad system may be used, wherein a new seed is used for each session (i.e. telephone call). In a hybrid system, the telephone and cellular carrier each have a reference noise key (e.g. 10,000 bits) from which the telephone selects a field of bits, such as 50 bits beginning at a randomly selected offset, and each uses this excerpt as the seed to generate the pseudo random data for encoding. Data sent from the telephone to the carrier (e.g. the offset) during call set-up allows the carrier to reconstruct the same pseudo random data for use in decoding. Yet further improvements can be derived by borrowing basic techniques from the art of cryptographic communications and applying them to the steganographically encoded signal detailed in this disclosure.

Details of applicant's preferred techniques for steganographic encoding/decoding with a pseudo random data stream are more particularly detailed in the previous portions of this specification, but the present technology is not limited to use with such techniques.

The reader is presumed to be familiar with cellular communications technologies. Accordingly, details known from prior art in this field aren't belabored herein.

Referring to Fig. 38, an illustrative cellular system includes a telephone 2010, a cell site 2012, and a central office 2014.

Conceptually, the telephone may be viewed as including a microphone 2016, an A/D converter 2018, a data formatter 2020, a modulator 2022, an RF section 2024, an antenna 2026, a demodulator 2028, a data unformatter 2030, a D/A converter 2032, and a speaker 2034.

In operation, a subscriber's voice is picked up by the microphone 2016 and converted to digital form by the A/D converter 2018. The data formatter 2020 puts the digitized voice into packet form, adding synchronization and control bits thereto. The modulator 2022 converts this digital data stream into an analog signal whose phase and/or amplitude properties change in accordance with the data being modulated. The RF section 2024 commonly translates this time-varying signal to one or more intermediate frequencies, and finally to a UHF transmission frequency. The RF section thereafter amplifies it and provides the resulting signal to the antenna 2026 for broadcast to the cell site 2012.

The process works in reverse when receiving. A broadcast from the cell cite is received through the antenna 2026. RF section 2024 amplifies and translates the received signal to a different frequency for demodulation. Demodulator 2028 processes the amplitude and/or phase variations of the signal provided by the RF section to produce a digital data stream corresponding thereto. The data unformatter 2030 segregates the voice data from the associated synchronization/control data, and passes the voice data to the D/A converter for conversion into analog form. The output from the D/A converter drives the speaker 2034, through which the subscriber hears the other party's voice.

The cell site 2012 receives broadcasts from a plurality of telephones 2010, and relays the data received to the central office 2014. Likewise, the cell site 2012 receives outgoing data from the central office and broadcasts same to the telephones.

The central office 2014 performs a variety of operations, including call authentication, switching, and cell hand-off.

(In some systems, the functional division between the cell site and the central station is different than that outlined above. Indeed, in some systems, all of this functionality is provided at a single site.)

In an exemplary embodiment of this aspect of the technology, each telephone 2010 additionally includes a steganographic encoder 2036. Likewise, each cell site 2012 includes a steganographic decoder 2038. The encoder operates to hide an auxiliary data signal among the signals representing the subscriber's voice. The decoder performs the reciprocal function, discerning the auxiliary data signal from the encoded voice signal. The auxiliary signal serves to verify the legitimacy of the call.

An exemplary steganographic encoder 2036 is shown in Fig. 39.

The illustrated encoder 2036 operates on digitized voice data, auxiliary data, and pseudo-random noise (PRN) data. The digitized voice data is applied at a port 2040 and is provided, e.g., from A/D converter 2018. The digitized voice may comprise 8-bit samples. The auxiliary data is applied at a port 2042 and comprises, in one form of the technology, a stream of binary data uniquely identifying the telephone 2010. (The auxiliary data may additionally include administrative data of the sort conventionally exchanged with a cell site at call set-up.) The pseudo-random noise data is applied at a port 2044 and can be, e.g., a signal that randomly alternates between "-1" and "1" values. (More and more cellular phones are incorporating spread spectrum capable circuitry, and this pseudo-random noise signal and other aspects of this technology can often "piggy-back" or share the circuitry which is already being applied in the basic operation of a cellular unit).

For expository convenience, it is assumed that all three data signals applied to the encoder 2036 are clocked at a common rate, although this is not necessary in practice.

In operation, the auxiliary data and PRN data streams are applied to the two inputs of a logic circuit 2046. The output of circuit 2046 switches between -1 and +1 in accordance with the following table:

AUX	PRN	OUTPUT
0	-1	1
0	1	-1
1	-1	-1
1	1	1

(If the auxiliary data signal is conceptualized as switching between -1 and 1, instead of 0 and 1, it will be seen that circuit 2046 operates as a one-bit multiplier.)

The output from gate 2046 is thus a bipolar data stream whose instantaneous value changes randomly in accordance with the corresponding values of the auxiliary data and the PRN data. It may be regarded as noise. However, it has the auxiliary data encoded therein. The auxiliary data can be extracted if the corresponding PRN data is known.

The noise-like signal from gate 2046 is applied to the input of a scaler circuit 2048. Scaler circuit scales (e.g. multiplies) this input signal by a factor set by a gain control circuit 2050. In the illustrated embodiment, this factor can range between 0 and 15. The output from scaler circuit 2048 can thus be represented as a five-bit data word (four bits, plus a sign bit) which changes each clock cycle, in accordance with the auxiliary and PRN data, and the scale

factor. The output from the scaler circuit may be regarded as "scaled noise data" (but again it is "noise" from which the auxiliary data can be recovered, given the PRN data).

The scaled noise data is summed with the digitized voice data by a summer 2051 to provide the encoded output signal (e.g. binarily added on a sample by sample basis). This output signal is a composite signal representing both the digitized voice data and the auxiliary data.

The gain control circuit 2050 controls the magnitude of the added scaled noise data so its addition to the digitized voice data does not noticeably degrade the voice data when converted to analog form and heard by a subscriber. The gain control circuit can operate in a variety of ways.

One is a logarithmic scaling function. Thus, for example, voice data samples having decimal values of 0, 1 or 2 may be correspond to scale factors of unity, or even zero, whereas voice data samples having values in excess of 200 may correspond to scale factors of 15. Generally speaking, the scale factors and the voice data values correspond by a square root relation. That is, a four-fold increase in a value of the voice data corresponds to approximately a two-fold increase in a value of the scaling factor associated therewith. Another scaling function would be linear as derived from the average power of the voice signal.

(The parenthetical reference to zero as a scaling factor alludes to cases, e.g., in which the digitized voice signal sample is essentially devoid of information content.)

More satisfactory than basing the instantaneous scaling factor on a single voice data sample, is to base the scaling factor on the dynamics of several samples. That is, a stream of digitized voice data which is changing rapidly can camouflage relatively more auxiliary data than a stream of digitized voice data which is changing slowly. Accordingly, the gain control circuit 2050 can be made responsive to the first, or preferably the second- or higher-order derivative of the voice data in setting the scaling factor.

In still other embodiments, the gain control block 2050 and scaler 2048 can be omitted entirely.

(Those skilled in the art will recognize the potential for "rail errors" in the foregoing systems. For example, if the digitized voice data consists of 8-bit samples, and the samples span the entire range from 0 to 255 (decimal), then the addition or subtraction of scaled noise to/from the input signal may produce output signals that cannot be represented by 8 bits (e.g. -2, or 257). A number of well-understood techniques exist to rectify this situation, some of them proactive and some of them reactive. Among these known techniques are: specifying that the digitized voice data shall not have samples in the range of 0-4 or 241-255, thereby safely permitting combination with the scaled noise signal; and including provision for detecting and adaptively modifying digitized voice samples that would otherwise cause rail errors.)

Returning to the telephone 2010, an encoder 2036 like that detailed above is desirably interposed between the A/D converter 2018 and the data formatter 2020, thereby serving to steganographically encode all voice transmissions with the auxiliary data. Moreover, the circuitry or software controlling operation of the telephone is arranged so that the auxiliary data is encoded repeatedly. That is, when all bits of the auxiliary data have been encoded, a pointer loops back and causes the auxiliary data to be applied to the encoder 2036 anew. (The auxiliary data may be stored at a known address in RAM memory for ease of reference.)

It will be recognized that the auxiliary data in the illustrated embodiment is transmitted at a rate one-eighth that of the voice data. That is, for every 8-bit sample of voice data, scaled noise data corresponding to a single bit of the auxiliary data is sent. Thus, if voice samples are sent at a rate of 4800 samples/second, auxiliary data can be sent at a rate of 4800 bits/second. If the auxiliary data is comprised of 8-bit symbols, auxiliary data can be conveyed at a rate of 600 symbols/second. If the auxiliary data consists of a string of even 60 symbols, each second of voice conveys the auxiliary data ten times. (Significantly higher auxiliary data rates can be achieved by resorting to more efficient coding techniques, such as limited-symbol codes (e.g. 5- or 6-bit codes), Huffman coding, etc.) This highly redundant transmission of the auxiliary data permits lower amplitude scaled noise data to be used while still providing sufficient signal-to-noise headroom to assure reliable decoding -- even in the relatively noisy environment associated with radio transmissions.

Turning now to Fig. 40, each cell site 2012 has a steganographic decoder 2038 by which it can analyze the composite data signal broadcast by the telephone 2010 to discern and separate the auxiliary data and digitized voice data therefrom. (The decoder desirably works on unformatted data (i.e. data with the packet overhead, control and administrative bits removed; this is not shown for clarity of illustration).

The decoding of an unknown embedded signal (i.e. the encoded auxiliary signal) from an unknown voice signal is best done by some form of statistical analysis of the composite data signal. The techniques therefor discussed above are equally applicable here. For example, the entropy-based approach can be utilized. In this case, the auxiliary data repeats every 480 bits (instead of every 8). As above, the entropy-based decoding process treats every 480th sample of the composite signal in like fashion. In particular, the process begins by adding to the 1st, 481st, 861st, etc. samples of the composite data signal the PRN data with which these samples were encoded. (That is, a set of sparse PRN data is added: the original PRN set, with all but every 480th datum zeroed out.) The localized entropy of the resulting signal around these points (i.e. the composite data signal with every 480th sample modified) is then computed.

The foregoing step is then repeated, this time subtracting the PRN data corresponding thereto from the 1st, 481st, 961st, etc. composite data samples.

One of these two operations will counteract (e.g. undo) the encoding process and reduce the resulting signal's entropy; the other will aggravate it. If adding the sparse PRN data to the composite data reduces its entropy, then this data must earlier have been subtracted from the original voice signal. This indicates that the corresponding bit of the auxiliary data signal was a "0" when these samples were encoded. (A "0" at the auxiliary data input of logic circuit 46 caused it to produce an inverted version of the corresponding PRN datum as its output datum, resulting in subtraction of the corresponding PRN datum from the voice signal.)

Conversely, if subtracting the sparse PRN data from the composite data reduces its entropy, then the encoding process must have earlier added this noise. This indicates that the value of the auxiliary data bit was a "1" when samples 1, 481, 961, etc., were encoded.

By noting in which case entropy is lower by (a) adding or (b) subtracting a sparse set of PRN data to/from the composite data, it can be determined whether the first bit of the auxiliary data is (a) a "0", or (b) a "1." (In real life applications, in the presence of various distorting phenomena, the composite signal may be sufficiently corrupted so that neither adding nor subtracting the sparse PRN data actually reduces entropy. Instead, both operations will increase entropy. In this case, the "correct" operation can be discerned by observing which operation increases the entropy less.)

The foregoing operations can then be conducted for the group of spaced samples of the composite data beginning with the second sample (i.e. 2, 482, 962, ...). The entropy of the resulting signals indicate whether the second bit of the auxiliary data signal is a "0" or a "1." Likewise with the following 478 groups of spaced samples in the composite signal, until all 480 bits of the code word have been discerned.

As discussed above, correlation between the composite data signal and the PRN data can also be used as a statistical detection technique. Such operations are facilitated in the present context since the auxiliary data whose encoded representation is sought, is known, at least in large part, a priori. (In one form of the technology, the auxiliary data is based on the authentication data exchanged at call set-up, which the cellular system has already received and logged; in another form (detailed below), the auxiliary data comprises a predetermined message.) Thus, the problem can be reduced to determining whether an expected signal is present or not (rather than looking for an entirely unknown signal). Moreover, data formatter 2020 breaks the composite data into frames of known length. (In a known GSM implementation, voice data is sent in time slots which convey 114 data bits each.) By padding the auxiliary data as necessary, each repetition of the auxiliary data can be made to start, e.g., at the beginning of such a frame of data. This, too, simplifies the correlation determinations, since 113 of every 114 possible bit alignments can be ignored (facilitating decoding even if none of the auxiliary data is known a priori).

Again, this wireless fraud detection poses the now-familiar problem of detecting known signals in noise, and the approaches discussed earlier are equally applicable here.

Where, as here, the location of the auxiliary signal is known a priori (or more accurately, known to fall within one of a few discrete locations, as discussed above), then the matched filter approach can often be reduced to a simple vector dot product between a set of sparse PRN data, and mean-removed excerpts of the composite signal corresponding thereto. (Note that the PRN data need not be sparse and may arrive in contiguous bursts, such as in British patent publication 2,196,167 mentioned earlier wherein a given bit in a message has contiguous PRN values associated with it.) Such a process steps through all 480 sparse sets of PRN data and performs corresponding dot product operations. If the dot product is positive, the corresponding bit of the auxiliary data signal is a "1;" if the dot product is negative, the corresponding bit of the auxiliary data signal is a "0." If several alignments of the auxiliary data signal within the framed composite signal are possible, this procedure is repeated at each candidate alignment, and the one yielding the highest correlation is taken as true. (Once the correct alignment is determined for a single bit of the auxiliary data signal, the alignment of all the other bits can be determined therefrom. "Alignment," perhaps better known as "synchronization," can be achieved by primarily through the very same mechanisms which lock on and track the voice signal itself and allow for the basic functioning of the cellular unit).

Security Considerations

Security of the presently described embodiments depends, in large part, on security of the PRN data and/or security of the auxiliary data. In the following discussion, a few of many possible techniques for assuring the security of these data are discussed.

In a first embodiment, each telephone 2010 is provided with a long noise key unique to the telephone. This key may be, e.g., a highly unique 10,000 bit string stored in ROM. (In most applications, keys substantially shorter than this may be used.)

The central office 2014 has access to a secure disk 2052 on which such key data for all authorized telephones are stored. (The disk may be remote from the office itself.)

Each time the telephone is used, fifty bits from this noise key are identified and used as the seed for a deterministic pseudo random number generator. The data generated by this PRN generator serve as the PRN data for that telephone call.

The fifty bit seed can be determined, e.g., by using a random number generator in the telephone to generate an offset address between 0 and 9,950 each time the telephone is used to place a call. The fifty bits in the noise key beginning at this offset address are used as the seed.

During call setup, this offset address is transmitted by the telephone, through the cell site 2012, to the central office 2014. There, a computer at the central office uses the offset address to index its copy of the noise key for that telephone. The central office thereby identifies the same 50 bit seed as was identified at the telephone. The central office 2014 then relays these 50 bits to the cell site 2012, where a deterministic noise generator like that in the telephone generates a PRN sequence corresponding to the 50 bit key and applies same to its decoder 2038.

By the foregoing process, the same sequence of PRN data is generated both at the telephone and at the cell site. Accordingly, the auxiliary data encoded on the voice data by the telephone can be securely transmitted to, and accurately decoded by, the cell site. If this auxiliary data does not match the expected auxiliary data (e.g. data transmitted at call set-up), the call is flagged as fraudulent and appropriate remedial action is taken.

It will be recognized that an eavesdropper listening to radio transmission of call set-up information can intercept only the randomly generated offset address transmitted by the telephone to the cell site. This data, alone, is useless in pirating calls. Even if the hacker had access to the signals provided from the central office to the cell site, this data too is essentially useless: all that is provided is a 50 bit seed. Since this seed is different for nearly each call (repeating only 1 out of every 9,950 calls), it too is unavailing to the hacker.

In a related system, the entire 10,000 bit noise key can be used as a seed. An offset address randomly generated by the telephone during call set-up can be used to identify where, in the PRN data resulting from that seed, the PRN data to be used for that session is to begin.

(Assuming 4800 voice samples per second, 4800 PRN data are required per second, or about 17 million PRN data per hour. Accordingly, the offset address in this variant embodiment will likely be far larger than the offset address described above.)

In this variant embodiment, the PRN data used for decoding is preferably generated at the central station from the 10,000 bit seed, and relayed to the cell site. (For security reasons, the 10,000 bit noise key should not leave the security of the central office.)

In variants of the foregoing systems, the offset address can be generated by the central station or at the cell site, and relayed to the telephone during call set-up, rather than vice versa.

In another embodiment, the telephone 2010 may be provided with a list of one-time seeds, matching a list of seeds stored on the secure disk 2052 at the central office. Each time the telephone is used to originate a new call, the next seed in the list is used. By this arrangement, no data needs to be exchanged relating to the seed; the telephone and the carrier each independently know which seed to use to generate the pseudo random data sequence for the current session.

In such an embodiment, the carrier can determine when the telephone has nearly exhausted its list of seeds, and can transmit a substitute list (e.g. as part of administrative data occasionally provided to the telephone). To enhance security, the carrier may require that the telephone be returned for manual reprogramming, to avoid radio transmission of this sensitive information. Alternatively, the substitute seed list can be encrypted for radio transmission using any of a variety of well known techniques.

In a second class of embodiments, security derives not from the security of the PRN data, but from security of the auxiliary message data encoded thereby. One such system relies on transmission of a randomly selected one of 256 possible messages.

In this embodiment, a ROM in the telephone stores 256 different messages (each message may be, e.g., 128 bits in length). When the telephone is operated to initiate a call, the telephone

randomly generates a number between 1 and 256, which serves as an index to these stored messages. This index is transmitted to the cell site during call set-up, allowing the central station to identify the expected message from a matching database on secure disk 2052 containing the same 256 messages. (Each telephone has a different collection of messages.) (Alternatively, the carrier may randomly select the index number during call set-up and transmit it to the telephone, identifying the message to be used during that session.) In a theoretically pure world where proposed attacks to a secure system are only mathematical in nature, much of these additional layers of security might seem superfluous. (The addition of these extra layers of security, such as differing the messages themselves, simply acknowledge that the designer of actual public-functioning secure systems will face certain implementation economics which might compromise the mathematical security of the core principles of this technology, and thus these auxiliary layers of security may afford new tools against the inevitable attacks on implementation).

Thereafter, all voice data transmitted by the telephone for the duration of that call is steganographically encoded with the indexed message. The cell site checks the data received from the telephone for the presence of the expected message. If the message is absent, or if a different message is decoded instead, the call is flagged as fraudulent and remedial action is taken.

In this second embodiment, the PRN data used for encoding and decoding can be as simple or complex as desired. A simple system may use the same PRN data for each call. Such data may be generated, e.g., by a deterministic PRN generator seeded with fixed data unique to the telephone and known also by the central station (e.g. a telephone identifier), or a universal noise sequence can be used (i.e. the same noise sequence can be used for all telephones). Or the pseudo random data can be generated by a deterministic PRN generator seeded with data that changes from call to call (e.g. based on data transmitted during call set-up identifying, e.g., the destination telephone number, etc.). Some embodiments may seed the pseudo random number generator with data from a preceding call (since this data is necessarily known to the telephone and the carrier, but is likely not known to pirates).

Naturally, elements from the foregoing two approaches can be combined in various ways, and supplemented by other features. The foregoing embodiments are exemplary only, and do not

begin to catalog the myriad approaches which may be used. Generally speaking, any data which is necessarily known or knowable by both the telephone and the cell site/central station, can be used as the basis for either the auxiliary message data, or the PRN data by which it is encoded.

Since the preferred embodiments of this aspect of the present technology each redundantly encodes the auxiliary data throughout the duration of the subscriber's digitized voice, the auxiliary data can be decoded from any brief sample of received audio. In the preferred forms of this aspect of the technology, the carrier repeatedly checks the steganographically encoded auxiliary data (e.g. every 10 seconds, or at random intervals) to assure that it continues to have the expected attributes.

While the foregoing discussion has focused on steganographically encoding a transmission from a cellular telephone, it will be recognized that transmissions to a cellular telephone can be steganographically encoded as well. Such arrangements find applicability, e.g., in conveying administrative data (i.e. non-voice data) from the carrier to individual telephones. This administrative data can be used, for example, to reprogram parameters of targeted cellular telephones (or all cellular telephones) from a central location, to update seed lists (for systems employing the above-described on-time pad system), to apprise "roaming" cellular telephones of data unique to an unfamiliar local area, etc.

In some embodiments, the carrier may steganographically transmit to the cellular telephone a seed which the cellular phone is to use in its transmissions to the carrier during the remainder of that session.

While the foregoing discussion has focused on steganographic encoding of the baseband digitized voice data, artisans will recognize that intermediate frequency signals (whether analog or digital) can likewise be steganographically encoded in accordance with principles of the technology. An advantage of post-baseband encoding is that the bandwidth of these intermediate signals is relatively large compared with the baseband signal, allowing more auxiliary data to be encoded therein, or allowing a fixed amount of auxiliary data to be repeated more frequently during transmission. (If steganographic encoding of an intermediate signal is employed, care

should be taken that the perturbations introduced by the encoding are not so large as to interfere with reliable transmission of the administrative data, taking into account any error correcting facilities supported by the packet format).

Those skilled in the art will recognize that the auxiliary data, itself, can be arranged in known ways to support error detecting, or error correcting capabilities by the decoder 38. The interested reader is referred, e.g., to Rorabaugh, *Error Coding Cookbook*, McGraw Hill, 1996, one of many readily available texts detailing such techniques.

While the preferred embodiment of this aspect of the technology is illustrated in the context of a cellular system utilizing packetized data, other wireless systems do not employ such conveniently framed data. In systems in which framing is not available as an aid to synchronization, synchronization marking can be achieved within the composite data signal by techniques such as that detailed in applicant's prior applications. In one class of such techniques, the auxiliary data itself has characteristics facilitating its synchronization. In another class of techniques, the auxiliary data modulates one or more embedded carrier patterns which are designed to facilitate alignment and detection.

As noted earlier, the principles of the technology are not restricted to use with the particular forms of steganographic encoding detailed above. Indeed, any steganographic encoding technique previously known, or hereafter invented, can be used in the fashion detailed above to enhance the security or functionality of cellular (or other wireless, e.g. PCS) communications systems. Likewise, these principles are not restricted to wireless telephones; any wireless transmission may be provided with an "in-band" channel of this type.

It will be recognized that systems for implementing applicant's technology can comprises dedicated hardware circuit elements, but more commonly comprise suitably programmed microprocessors with associated RAM and ROM memory (e.g. one such system in each of the telephone 2010, cell-site 2012, and central office 2014).

ENCODING BY BIT CELLS

The foregoing discussions have focused on incrementing or decrementing the values of individual pixels, or of groups of pixels (bumps), to reflect encoding of an auxiliary data signal combined with a pseudo random noise signal. The following discussion details a variant embodiment wherein the auxiliary data -- without pseudo randomization -- is encoded by patterned groups of pixels, here termed bit cells.

Referring to Figs 41A and 41B, two illustrative 2x2 bit cells are shown. Fig. 41A is used to represent a "0" bit of the auxiliary data, while Fig. 41B is used to represent a "1" bit. In operation, the pixels of the underlying image are tweaked up or down in accordance with the +/- values of the bit cells to represent one of these two bit values. (The magnitude of the tweaking at any given pixel, bit cell or region of the image can be a function of many factors, as detailed below. It is the sign of the tweaking that defines the characteristic pattern.) In decoding, the relative biases of the encoded pixels are examined (using techniques described above) to identify, for each corresponding region of the encoded image, which of the two patterns is represented.

While the auxiliary data is not explicitly randomized in this embodiment, it will be recognized that the bit cell patterns may be viewed as a "designed" carrier signal, as discussed above.

The substitution of the previous embodiments' pseudo random noise with the present "designed" information carrier affords an advantage: the bit cell patterning manifests itself in Fourier space. Thus, the bit cell patterning can act like the subliminal digital graticules discussed above to help register a suspect image to remove scale/rotation errors. By changing the size of the bit cell, and the pattern therein, the location of the energy thereby produced in the spatial transform domain can be tailored to optimize independence from typical imagery energy and facilitate detection.

(While the foregoing discussion contemplates that the auxiliary data is encoded directly -- without randomization by a PRN signal, in other embodiments, randomization can of course be used.)

MORE ON PERCEPTUALLY ADAPTIVE SIGNING

In several of the above-detailed embodiments, the magnitude of the signature energy was tailored on a region-by-region basis to reduce its visibility in an image (or its audibility in a sound, etc.). In the following discussion, applicant more particularly considers the issue of hiding signature energy in an image, the separate issues thereby posed, and solutions to each of these issues.

The goal of the signing process, beyond simply functioning, is to maximize the "numeric detectability" of an embedded signature while meeting some form of fixed "visibility/acceptability threshold" set by a given user/creator.

In service to design toward this goal, imagine the following three axis parameter space, where two of the axes are only half-axes (positive only), and the third is a full axis (both negative and positive). This set of axes define two of the usual eight octal spaces of euclidean 3-space. As things refine and "deservedly separable" parameters show up on the scene (such as "extended local visibility metrics"), then they can define their own (generally) half-axis and extend the following example beyond three dimensions.

The signing design goal becomes optimally assigning a "gain" to a local bump based on its coordinates in the above defined space, whilst keeping in mind the basic needs of doing the operations fast in real applications. To begin with, the three axes are the following. We'll call the two half axes x and y, while the full axis will be z.

The x axis represents the luminance of the singular bump. The basic idea is that you can squeeze a little more energy into bright regions as opposed to dim ones. It is important to note that when true "psycho-linear - device independent" luminance values (pixel DN's) come along, this axis might become superfluous, unless of course if the luminance value couples into the other operative axes (e.g. C*xy). For now, this is here as much due to the sub-optimality of current quasi-linear luminance coding.

The y axis is the "local hiding potential" of the neighborhood within which the bump finds itself. The basic idea is that flat regions have a low hiding potential since the eye can detect subtle changes in such regions, whereas complex textured regions have a high hiding potential. Long lines and long edges tend toward the lower hiding potential since "breaks and choppiness" in nice smooth long lines are also somewhat visible, while shorter lines and edges, and mosaics thereof, tend toward the higher hiding potential. These latter notions of long and short are directly connected to processing time issues, as well to issues of the engineering resources needed to carefully quantify such parameters. Developing the working model of the y-axis will inevitably entail one part theory to one part picky-artist-empiricism. As the parts of the hodge-podge y-axis become better known, they can splinter off into their own independent axes if it's worth it.

The z-axis is the "with or against the grain" (discussed below) axis which is the full axis - as opposed to the other two half-axes. The basic idea is that a given input bump has a pre-existing bias relative to whether one wishes to encode a '1' or a '0' at its location, which to some non-trivial extent is a function of the reading algorithms which will be employed, whose (bias) magnitude is semi-correlated to the "hiding potential" of the y-axis, and, fortunately, can be used advantageously as a variable in determining what magnitude of a tweak value is assigned to the bump in question. The concomitant basic idea is that when a bump is already your friend (i.e. its bias relative to its neighbors already tends towards the desired delta value), then don't change it much. Its natural state already provides the delta energy needed for decoding, without altering the localized image value much, if at all. Conversely, if a bump is initially your enemy (i.e. its bias relative to its neighbors tends away from the delta sought to be imposed by the encoding), then change it an exaggerated amount. This later operation tends to reduce the excursion of this point relative to its neighbors, making the point less visibly conspicuous (a

highly localized blurring operation), while providing additional energy detectable when decoding. These two cases are termed "with the grain" and "against the grain" herein.

The above general description of the problem should suffice for many years. Clearly adding in chrominance issues will expand the definitions a bit, leading to a bit more signature bang for the visibility, and human visibility research which is applied to the problem of compression can equally be applied to this area but for diametrically opposed reasons. Here are guiding principles which can be employed in an exemplary application.

For speed's sake, local hiding potential can be calculated only based on a 3 by 3 neighborhood of pixels, the center one being signed and its eight neighbors. Beyond speed issues, there is also no data or coherent theory to support anything larger as well. The design issue boils down to canning the y-axis visibility thing, how to couple the luminance into this, and a little bit on the friend/enemy asymmetry thing. A guiding principle is to simply make a flat region zero, a classic pure maxima or minima region a "1.0" or the highest value, and to have "local lines", "smooth slopes", "saddle points" and whatnot fall out somewhere in between.

The exemplary application uses six basic parameters: 1) luminance; 2) difference from local average; 3) the asymmetry factor (with or against the grain); 4) minimum linear factor (our crude attempt at flat v. lines v. maxima); 5) bit plane bias factor; and 6) global gain (the user's single top level gain knob).

The Luminance, and Difference from Local Average parameters are straight forward, and their use is addressed elsewhere in this specification.

The Asymmetry factor is a single scalar applied to the "against the grain" side of the difference axis of number 2 directly above.

The Minimum Linear factor is admittedly crude but it should be of some service even in a 3 by 3 neighborhood setting. The idea is that true 2D local minima and maxima will be highly perturbed along each of the four lines travelling through the center pixel of the 3 by 3

neighborhood, while a visual line or edge will tend to flatten out at least one of the four linear profiles. [The four linear profiles are each 3 pixels in length, i.e., the top left pixel - center - bottom right; the top center - center - bottom center; the top right - center - bottom left; the right center - center - left center;]. Let's choose some metric of entropy as applied to three pixels in a row, perform this on all four linear profiles, then choose the minimum value for our ultimate parameter to be used as our 'y-axis.'

The Bit Plane Bias factor is an interesting creature with two faces, the pre-emptive face and the post-emptive face. In the former, you simply "read" the unsigned image and see where all the biases fall out for all the bit planes, then simply boost the "global gain" of the bit planes which are, in total, going against your desired message, and leave the others alone or even slightly lower their gain. In the post-emptive case, you churn out the whole signing process replete with the pre-emptive bit plane bias and the other 5 parameters listed here, and then you, e.g., run the signed image through heavy JPEG compression AND model the "gestalt distortion" of line screen printing and subsequent scanning of the image, and then you read the image and find out which bit planes are struggling or even in error, you appropriately beef up the bit plane bias, and you run through the process again. If you have good data driving the beefing process you should only need to perform this step once, or, you can easily Van-Cittertize the process (arcane reference to reiterate the process with some damping factor applied to the tweaks).

Finally, there is the Global Gain. The goal is to make this single variable the top level "intensity knob" (more typically a slider or other control on a graphical user interface) that the slightly curious user can adjust if they want to. The very curious user can navigate down advanced menus to get their experimental hands on the other five variables here, as well as others.

Visible Watermark

In certain applications it is desirable to apply a visible indicia to an image to indicate that it includes steganographically encoded data. In one embodiment, this indicia can be a lightly visible logo (sometimes termed a "watermark") applied to one corner of the image. This indicates that the image is a "smart" image, conveying data in addition to the imagery. A lightbulb is one suitable logo.

If marking of images becomes widespread (e.g., by software compatible with Adobe's image processing software), a user of such software can decode the embedded data from an image and consult a public registry to identify the proprietor of the image. In some embodiments, the registry can serve as the conduit through which appropriate royalty payments are forwarded to the proprietor for the user's use of an image. (In an illustrative embodiment, the registry is a server on the Internet, accessible via the World Wide Web, coupled to a database. The database includes detailed information on catalogued images (e.g. name, address, phone number of

proprietor, and a schedule of charges for different types of uses to which the image may be put), indexed by identification codes with which the images themselves are encoded. A person who decodes an image queries the registry with the codes thereby gleaned to obtain the desired data and, if appropriate, to forward electronic payment of a copyright royalty to the image's proprietor.)

Particular Data Formats

While the foregoing steganography techniques are broadly applicable, their commercial acceptance will be aided by establishment of standards setting forth which pixels/bit cells represent what. The following discussion proposes one set of possible standards. For expository convenience, this discussion focuses on decoding of the data; encoding follows in a straightforward manner.

Referring to Fig. 42, an image 1202 includes a plurality of tiled "signature blocks" 1204. (Partial signature blocks may be present at the image edges.) Each signature block 1204 includes an 8 x 8 array of sub-blocks 1206. Each sub-block 1206 includes an 8 x 8 array of bit cells 1208. Each bit cell comprises a 2 x 2 array of "bumps" 1210. Each bump 1210, in turn, comprises a square grouping of 16 individual pixels 1212.

The individual pixels 1212 are the smallest quanta of image data. In this arrangement, however, pixel values are not, individually, the data carrying elements. Instead, this role is served by bit cells 1208 (i.e. 2 x 2 arrays of bumps 1210). In particular, the bumps comprising the bits cells are encoded to assume one of the two patterns shown in Fig. 41. As noted earlier, the pattern shown in Fig. 41A represents a "0" bit, while the pattern shown in Fig. 41B represents a "1" bit. Each bit cell 1208 (64 image pixels) thus represents a single bit of the embedded data. Each sub-block 1206 includes 64 bit cells, and thus conveys 64 bits of embedded data.

(The nature of the image changes effected by the encoding follows the techniques set forth above under the heading MORE ON PERCEPTUALLY ADAPTIVE SIGNING; that discussion is not repeated here.)

In the illustrated embodiment, the embedded data includes two parts: control bits and message bits. The 16 bit cells 1208A in the center of each sub-block 1206 serve to convey 16 control bits. The surrounding 48 bit cells 1208B serve to convey 48 message bits. This 64-bit chunk of data is encoded in each of the sub-blocks 1206, and is repeated 64 times in each signature block 1204.

A digression: in addition to encoding of the image to redundantly embed the 64 control/message bits therein, the values of individual pixels are additionally adjusted to effect encoding of subliminal graticules through the image. In this embodiment, the graticules discussed in conjunction with Fig. 29A are used, resulting in an imperceptible texturing of the image. When the image is to be decoded, the image is transformed into the spatial domain, the Fourier-Mellin technique is applied to match the graticule energy points with their expected positions, and the processed data is then inverse-transformed, providing a registered image ready for decoding. (The sequence of first tweaking the image to effect encoding of the subliminal graticules, or first tweaking the image to effect encoding of the embedded data, is not believed to be critical. As presently practiced, the local gain factors (discussed above) are computed; then the data is encoded; then the subliminal graticule encoding is performed. (Both of these encoding steps make use of the local gain factors.))

Returning to the data format, once the encoded image has been thus registered, the locations of the control bits in sub-block 1206 are known. The image is then analyzed, in the aggregate (i.e. considering the "northwestern-most" sub-block 1206 from each signature block 1204), to determine the value of control bit #1 (represented in sub-block 1206 by bit cell 1208Aa). If this value is determined (e.g. by statistical techniques of the sort detailed above) to be a "1," this indicates that the format of the embedded data conforms to the standard detailed herein (the Digimarc Beta Data Format). According to this standard, control bit #2 (represented by bit cells 1208Ab) is a flag indicating whether the image is copyrighted. Control bit #3

(represented by bit cells 1208Ac) is a flag indicating whether the image is unsuitable for viewing by children. Certain of the remaining bits are used for error detection/correction purposes.

The 48 message bits of each sub block 1206 can be put to any use; they are not specified in this format. One possible use is to define a numeric "owner" field and a numeric "image/item" field (e.g. 24 bits each).

If this data format is used, each sub-block 1206 contains the entire control/message data, so same is repeated 64 times within each signature block of the image.

If control bit #1 is not a "1," then the format of the embedded data does not conform to the above described standard. In this case, the reading software analyzes the image data to determine the value of control bit #4. If this bit is set (i.e. equal to "1"), this signifies an embedded ASCII message. The reading software then examines control bits #5 and #6 to determine the length of the embedded ASCII message.

If control bits #5 and #6 both are "0," this indicates the ASCII message is 6 characters in length. In this case, the 48 bit cells 1208B surrounding the control bits 1208A are interpreted as six ASCII characters (8 bits each). Again, each sub-block 1206 contains the entire control/message data, so same is repeated 64 times within each signature block 1204 of the image.

If control bit #5 is "0" and control bit #6 is "1," this indicates the embedded ASCII message is 14 characters in length. In this case, the 48 bit cells 1208B surrounding the control bits 1208A are interpreted as the first six ASCII characters. The 64 bit cells 1208 of the immediately-adjoining sub-block 1220 are interpreted as the final eight ASCII characters.

Note that in this arrangement, the bit-cells 1208 in the center of sub-block 1220 are not interpreted as control bits. Instead, the entire sub-block serves to convey additional message bits. In this case there is just one group of control bits for two sub-blocks.

Also note than in this arrangement, pairs of sub-blocks 1206 contains the entire control/message data. so same is repeated 32 times within each signature block 1204 of the image.

Likewise if control bit #5 is "1" and control bit #6 is "0." This indicates the embedded ASCII message is 30 characters in length. In this case, 2 x 2 arrays of sub-blocks are used for each representation of the data. The 48 bit cells 1208B surrounding control bits 1208A are interpreted as the first six ASCII characters. The 64 bit cells of each of adjoining block 1220 are interpreted as representing the next 8 additional characters. The 64 bits cells of sub-block 1222 are interpreted as representing the next 8 characters. And the 64 bit cells of sub-block 1224 are interpreted as representing the final 8 characters. In this case, there is just one group of control bits for four sub-blocks. And the control/message data is repeated 16 times within each signature block 1204 of the image.

If control bits #5 and #6 are both "1"s, this indicates an ASCII message of programmable length. In this case, the reading software examines the first 16 bit cells 1208B surrounding the control bits. Instead of interpreting these bit cells as message bits, they are interpreted as additional control bits (the opposite of the case described above, where bit cells normally used to represent control bits represented message bits instead). In particular, the reading software interprets these 16 bits as representing, in binary, the length of the ASCII message. An algorithm is then applied to this data (matching a similar algorithm used during the encoding process) to establish a corresponding tiling pattern (i.e. to specify which sub-blocks convey which bits of the ASCII message, and which convey control bits.)

In this programmable-length ASCII message case, control bits are desirably repeated several times within a single representation of the message so that, e.g., there is one set of control bits for approximately every 24 ASCII characters. To increase packing efficiency, the tiling algorithm can allocate (divide) a sub-block so that some of its bit-cells are used for a first representation of the message, and others are used for another representation of the message.

Reference was earlier made to beginning the decoding of the registered image by considering the "northwestern-most" sub-block 1206 in each signature block 1204. This bears elaboration.

Depending on the data format used, some of the sub-blocks 1206 in each signature block 1204 may not include control bits. Accordingly, the decoding software desirably determines the data format by first examining the "northwestern-most" sub-block 1206 in each signature block 1204; the 16 bits cells in the centers of these sub-blocks will reliably represent control bits. Based on the value(s) of one or more of these bits (e.g. the Digimarc Beta Data Format bit), the decoding software can identify all other locations throughout each signature block 1204 where the control bits are also encoded (e.g. at the center of each of the 64 sub-blocks 1206 comprising a signature block 1204), and can use the larger statistical base of data thereby provided to extract the remaining control bits from the image (and to confirm, if desired, the earlier control bit(s) determination). After all control bits have thereby been discerned, the decoding software determines (from the control bits) the mapping of message bits to bit cells throughout the image.

To reduce the likelihood of visual artifacts, the numbering of bit cells within sub-blocks is alternated in a checkerboard-like fashion. That is, the "northwestern-most" bit cell in the "northwestern-most" sub-block is numbered "0." Numbering increases left to right, and successively through the rows, up to bit cell 63. Each sub-block diametrically adjoining one of its corners (i.e. sub-block 1224) has the same ordering of bit cells. But sub-blocks adjoining its edges (i.e. sub-blocks 1220 and 1222) have the opposite numbering. That is, the "northwestern-most" bit cell in sub-blocks 1220 and 1222 is numbered "63." Numbering decreases left to right, and successively through the rows, down to 0. Likewise throughout each signature block 1204.

In a variant of the Digimarc beta format, a pair of sub-blocks is used for each representation of the data, providing 128 bit cells. The center 16 bit cells 1208 in the first sub-block 1206 are used to represent control bits. The 48 remaining bit cells in that sub-block, together with all 64 bit cells 1208 in the adjoining sub-block 1220, are used to provide a 112-bit message field. Likewise for every pair of sub-blocks throughout each signature block 1204. In such an arrangement, each signature block 1204 thus includes 32 complete representations of the

encoded data (as opposed to 64 representations in the earlier-described standard). This additional length allows encoding of longer data strings, such as a numeric IP address (e.g. URL).

Obviously, numerous alternative data formats can be designed. The particular format used can be indicated to the decoding software by values of one or more control bits in the encoded image.

Other Applications

Before concluding, it may be instructive to review some of the other fields where principles of applicant's technology can be employed.

One is smart business cards, wherein a business card is provided with a photograph having unobtrusive, machine-readable contact data embedded therein. (The same function can be achieved by changing the surface microtopology of the card to embed the data therein.)

Another promising application is in content regulation. Television signals, images on the internet, and other content sources (audio. image, video, etc.) can have data indicating their "appropriateness" (i.e. their rating for sex, violence, suitability for children, etc.) actually embedded in the content itself rather than externally associated therewith. Television receivers, internet surfing software, etc., can discern such appropriateness ratings (e.g. by use of universal

code decoding) and can take appropriate action (e.g. not permitting viewing of an image or video, or play-back of an audio source).

In a simple embodiment of the foregoing, the embedded data can have one or more "flag" bits, as discussed earlier. One flag bit can signify "inappropriate for children." (Others can be, e.g., "this image is copyrighted" or "this image is in the public domain.") Such flag bits can be in a field of control bits distinct from an embedded message, or can -- themselves -- be the message. By examining the state of these flag bits, the decoder software can quickly apprise the user of various attributes of the image.

(As discussed earlier, control bits can be encoded in known locations in the image -- known relative to the subliminal graticules -- and can indicate the format of the embedded data (e.g. its length, its type, etc.) As such, these control bits are analogous to data sometimes conveyed in prior art file headers, but in this case they are embedded within an image, instead of prepended to a file.)

The field of merchandise marking is generally well served by familiar bar codes and universal product codes. However, in certain applications, such bar codes are undesirable (e.g. for aesthetic considerations, or where security is a concern). In such applications, applicant's technology may be used to mark merchandise, either through in innocuous carrier (e.g. a photograph associated with the product), or by encoding the microtopology of the merchandise's surface, or a label thereon.

There are applications -- too numerous to detail -- in which steganography can advantageously be combined with encryption and/or digital signature technology to provide enhanced security.

Medical records appear to be an area in which authentication is important. Steganographic principles -- applied either to film-based records or to the microtopology of documents -- can be employed to provide some protection against tampering.

Many industries, e.g. automobile and airline, rely on tags to mark critical parts. Such tags, however, are easily removed, and can often be counterfeited. In applications wherein better security is desired, industrial parts can be steganographically marked to provide an inconspicuous identification/authentication tag.

In various of the applications reviewed in this specification, different messages can be steganographically conveyed by different regions of an image (e.g. different regions of an image can provide different internet URLs, or different regions of a photocollage can identify different photographers). Likewise with other media (e.g. sound).

Some software visionaries look to the day when data blobs will roam the datawaves and interact with other data blobs. In such an era, it will be necessary for such blobs to have robust and incorruptible ways of identifying themselves. Steganographic techniques again hold much promise here.

Finally, message changing codes -- recursive systems in which steganographically encoded messages actually change underlying steganographic code patterns -- offer new levels of sophistication and security. Such message changing codes are particularly well suited to applications such as plastic cash cards where time-changing elements are important to enhance security.

Again, while applicant prefers the particular forms of steganographic encoding detailed above, the diverse applications disclosed in this specification can largely be practiced with other steganographic marking techniques, many of which are known in the prior art. Likewise, while the specification has focused on applications of this technology to images, the principles thereof are generally equally applicable to embedding such information in audio, physical media, or any other carrier of information.

Finally, while the specification has been illustrated with particular embodiments, it will be recognized that elements, components and steps from these embodiments can be recombined in

different arrangements to serve different needs and applications, as will be readily apparent to those of ordinary skill in the art.

POINTER TO A HEADER, AND/OR, POINTER TO A PROCESS

The basic philosophy surrounding the use of a "header" in a digital file has already been discussed, along with the idea that steganographically encoded information can serve many of these functions independently or in parallel with a traditional header. This section attempts to further clarify this notion as well as further clarify the notion that attached information can act as a "pointer" to a remote database containing the information that might be otherwise contained in a header. Refer to the sections on header verification, the "bodier", and network linking methods and hot objects in general. This section also expands upon the idea that header information and other kinds of attached information can be more than simply passive descriptive information, but can be fully executable code or other forms of "active information and network links."

As previously developed in the section on hot objects, given that realtime connection of a desktop computer, a television set or a stero system onto a network is becoming more and more commonplace, the absolute need for an attached header replete with full information about a data object is diminishing, and the possibility of simply pointing to a remote yet connected database is becoming more and more practical. The persistent and immediate connection to the information is more fundamental than strictly needing the full information attached to the data object (where time to change information presents a particular problem for object-attached information).

Referring back to Fig. 16, and harkening back to the discussion on networked linking methods, we find the situation depicted in Fig. 60. The header information "joe's image" is now only located on some accessible remote database, and the information attached to the object in either the actual header, or steganographically embedded, or both, now contains simply an address to the remote database location. Software applications would then be manufactured such that a networked connection would be assumed to be either active or capable of becoming so rather swiftly, and that

when a data object such as an image or a video clip was read in, the embedded address of the header information would be read, a quick retrieve of the database information would be performed, and the remote header information would be used just as if it were locally attached to the data object.

It is important to note that the stegonographic embedded information does not necessarily need to be present in order to make such a system function, and vice versa, the traditional header need not be present if steganographic information is present. The two together represent a value-added redundancy and greater assurance that the link between the remote information and the object remain intact. Likewise, even if there does exist some "master remote header" for a given data object, an "static-in-time copy" (by definition) of that master header file can still travel with the data object for the purposes of speed or for the purposes of servicing non-networked sties or applications, with reference back to the master if need be for any latest updates. Likewise, though the principles of this application focus largely on audio-visual signals and data objects, many of these design principles apply to text files, software programs, basic data records, and the like, where steganography per se may not even be applicable. Likewise, any other kind of information can certainly still be attached to an object in addition to the above mentioned "header" information. Another way to look at it is that this new "header address" can simply be appended to or incorporated into any existing header format such as JPEG, TIFF, etc.

It is also important to note that the information stored remotely need not simply be "passive information" such as a person's name or the latest price for some peddled item. It can also be computer instructions or software executable code that is associated with the data object. Thus a software application can read in a data object, reading at the same time the remote database address, retrieve via a network the remote connection, then either display the remote information along with the data object information itself, or actually launch some local application much like JAVA based systems currently function, or the remote database can actually ship executable code which may then be invoked locally on the machine where the data object has been read. In this fashion, a data object becomes a first stepping stone into some computing experience usually related to the content of the object itself, much as a cover of a book is a quick visual summary of the book's content where the challenge becomes enticing

third parties to enter.

The notion of a singular string of attached bits referring to a remote database wherein a master header is located we shall henceforth refer to as an "identity pointer". We shall refer to the database location containing the master information connected with that data object as the "master identity."

As is well known even in the prior art of traditional mail, addresses can be hierarchical in nature, with ever finer levels of granularity in searching for the ultimate central location or master identity. So too with identity pointers, they can actually be a single string of hierarchical information parsed according to where an application or machine needs to go in order to retrieve the full information associated with the data object and/or audio/visual signal. Thus, a sub-string inside the identity pointer might indicate, through proper programming of an application or instrument including address redirection tables, that the master identity for such and such object exists inside one's own computer at such and such directory and in such and such file. On the other hand, a first section of the identity pointer might indicate a search is required on the world wide web for a particular rights management database or commerce server database, a second section of the identity pointer then "points" to an address at this particular database, wherein what is pointed to may be the ultimate location of the master identity, or the pointed-to location is actually a database of further pointers to a multitude of databases, indicating that a third section of the identity pointer now has an address location of the ultimate master identity. This formula can clearly recurse further into fourth, fifth, and so on layers. The point is that it is not strictly required to have the identity pointer only point to some singular master database, but that there will in fact be a multitude of interconnected databases which will be the rule rather than the exception. This includes databases which are more traditionally viewed as belonging in some kind of network which is not necessarily as large as the internet, such as local area networks, or even a personal computers series of hard drives, or whatever. The main overall point is that the identity pointer contains sufficient information in order to be able to find the master identity, where it is a given that an application or instrument can read the master pointer, interpret it, and do the searching and retrieving of the master identity.

Those skilled in the art may notice that so far in this section no mention has been

made about encrypting either the identity pointer or the master identity or any communications involved in the overall process. Such improvements can certainly be made and enhance more of the "secure identification" aspects of an attached information system, or the related "verification" function that many commerce applications will be seeking. The basic principles of the invention can nevertheless be followed.

The identity pointer can also point to locations which are not (currently) viewed as strictly a database. For example, a data object can contain a sub-string which points to a particular application that needs to be launched in order to resolve the remainder of the identity pointer string. A more specific example of the general example might be that a sub-string in the identity pointer instructs a machine to launch a web browser, and the remainder of the string would be a traditional URL to some web page containing basic contact information to the creator of a video production. Users on a local machine might program the machine to their own liking in that they would "allow" content with these abilities to go ahead and launch browsers when instructed, or they can tell their local machine to ignore these requests by a given piece of content. Likewise, an sub-string in the identity pointer might instruct a multimedia database interface program to be launched and the location of the object in question sought, found, and displayed.

It can be readily seen that the general principles of the invention would indicate that applications and instruments which deal with data objects require less and less local intelligence regarding "interpreting and operating upon" attached information, yet instead they become better and better at being realtime connected to a broader network and know how to properly find and retrieve information from various locations on that broader network. Moreover, applications and instruments so enabled would also make use of well-known-in-the-art methods of launching new execution threads or entire new applications based on the retrieved information. The physical location of the master identity, as well as the "level" it exists on in the network hierarchy (from CPU RAM, through Operating System, through local storage disks, through local networks, up through global networks such as the Internet) becomes immaterial so long as an application or instrument is equipped with the realtime ability to find and retrieve the pertinent information.

We now turn toward a more detailed description of the drawings. Fig. 60 depicts

the essential concept behind the "master identity object" and a system built around it. ("Master Identity Object", or MIO, is here used as a convenience and should not be construed as limiting the kind of information pointed to by an embedded pointer; i.e., the MIO could easily point to executable code or executable sequences for example rather than identity information). Indeed, the system is essential to the definition of the object. Quite simply, a header is replaced by a pointer to a header, and the header information itself is located in some accessible location. The pointer 1690 represents whatever complete address information is needed in order for an instrument or application to locate the header information, 1694, located in some binary storage location (or database) 1692. The data stream 802 is whatever normal data that might be contained in a typical digital image file or binary record or software program or whatever. The pointer, 1690, can also be embedded steganographically as depicted, or alternatively, it can ONLY be associated with the data stream steganographically.

Fig. 61 begins to dissect Fig. 60, trying to depict a closer view of a practical MIO ("master identity object"). Reverence is paid to the idea that common data objects and common data formats will be used for some time to come, and in many ways the MIO information will be appended. Thus we start with a traditional header 1700 and the associated data stream 802. The MIO pointer is now split out into three pieces, a "tag" which identifies the following information as a pointer to a remote database. The pointer is now conveniently split into a pointer to a pointer 1704, effectively giving the address of some database in a sea of potential databases, while pointer 1706 is the ultimate address on the found database pointed to by 1704. Indeed, 1704 could also be split out, just as in surface mail type addressing.

Fig. 61 also indicates that multiple layers of MIO information can be added to an object. One example of this would be that an organization would track and account for all incoming and outgoing data objects and wish to provide a pointer to that given objects lifetime within the organizations data network, yet respect any previous MIO information that the creator or distributor of a data object already attached to that object. There are numerous other examples, including derivative work applications or situations where a data object evolves.

Fig. 62 takes a closer look at the connectivity components of an MIO system, looked at through the eyes of an instrument or application needing to retrieve the master

identity information for a given MIO-enabled data object. An arbitrary data object 1710 enters a highly genericized "node" 1712. Typically, this node might be an "open file" procedure in a software application, or it might be a "search for all image files" operation initiated in a computer's operating system, or it might be a digital receiver in a television set.

The node 1712 is meant to represent the place where a given data object enters a given instrument or application and is scanned for MIO traits. The node 1712's output is simply that the data object either is or is not an MIO type object. If it discovers an MIO object as in 1716, it must next ask the simple question of whether or not the given instruments cares or not, essentially asking whether or not to go fetch the information pointed to by the embedded pointer. If it is not configured to do so, it goes on with its work, 1718. If it is configured to go fetch the associated remote information (1720), it then simply goes through a basic address resolution process to locate the master identity. A first low level search might be typical wherein the "master identity" (which we have amply explained might actually be a search for further instructions rather than only "looking for a name") is potentially found on a user's own desktop computer or local instrument. This might be as low level as a call to some memory location in the CPU system 1724, perhaps retrieving and therefor instigating instructions to prepare for video information to follow. Or it might be a pointer 1726 to some location in a display device, perhaps as a precursor to understanding the current display settings that a user is operating under.

The need for multiple layers of MIO information as depicted in Fig. 61 becomes more apparent in these examples as well, in that the lower level retrieving will often be precursors to more advanced actions. Likewise, an address might be found in the operating system 1728, ROM, RAM, hard drives, etc. (1730). A token indication of a firewall 1732 is depicted showing that some addresses will only be accessible to a "local object" by design. Finally, perhaps the more common use of true "identity" will come from a route to a broader network 1734. Depicted is a certain remote database 1736 where the given data object might have its master identity. These two basic ideas of "local" identity and "remote" identity are depicted below as 1738 and 1740, but it should be appreciated that these can become quite blurry in their distinctions.

It should be understood that one object of this invention is that, for many

practical systems, the locating process of finding and retrieving the master identity information or instructions is meant to take place in realtime, that is, is typically less than a second or two ideally, and no more than ten seconds or twenty seconds even in case where there is in the multiples of kilobytes to retrieve. This kind of time performance is generally possible today on the internet even with sub-100Kbit per second modems, though less than a second is generally out of reach of current modem-based systems using standard phone lines and standard internet communications protocols.

Fig. 63 traces out a generic process flow within an instrument or application with a slightly different twist than depicted in Fig. 62. Again, any device capable of processing a data object reads said object in, 1750, and asks the simple question of whether or not remote associated information exists for that object 1752. If not, fine, 1754, go about your way. If so, it asks should I care, 1758, if not (1760, go about your way. If so, 1762, go fetch the information (or series of information if there is recursive MIO information). Box 1766 then explicitly shows that the user's device will still effect what happens even after the remote data is retrieved.

The permutations and combinations of operations from this point forward are as broad as general computing principles permit. As one tiny example, the user may have configured the device such that if the retrieved information does not match certain "key words" that the user has chosen, then don't even bother displaying an image (assuming that is the data object in question). Essentially, the retrieved information from 1764 and the user configuration 1766 set up a generally unlimited set of new instructions to the device. Another example might be that the user configuration 1766 says to go ahead and follow whatever instructions come back with the retrieved information, such as "launch a browser and follow an URL to some JAVA web site and don't mind sending out my credit card information if they ask". This is deliberately melodramatic to illustrate that the basic branching at 1770 is as broad as general computing principles permit.

Fig. 64 briefly illustrates that both embedded information as well as pointed-to information can actually sequences of instructions, including decision branched instructions, not merely singular instructions or singular blocks of passive descriptive information. This idea is encapsulated in the word "script" as is generally well known

in the art of computing.

Fig. 65 essentially depicts an MIO system operated in "reverse" from that of the previous few figures. In general, the idea here is that the master identity wants to know and collect all migration activities of its associated data object, and the system depicted in Fig. 65 is essentially a way to do this. Again, we see a generic data object 1820 entering some generic application or instrument 1822, here referred to as a "reporting node". The first question still is "are you an MIO?" 1824. 1826 alludes to the idea of reporting non-hits to some kind of master MIO system database 1830, a database which might track the broad traffic patterns of MIO object vs. Non-MIO objects. If the node finds an MIO, the next logical question is "may" I send a message to its master identity in order to inform the master identity that this object passed this location at this particular time? (1834). This is important since there will be many situations due to privacy and security considerations where such reporting would be intolerable 1834. However, there may also be many situations where this reporting information is valuable to all users of MIO based systems, and a "count" message will be sent to the master identity location, where locating the master identity follows previous descriptions. Details of how the master identity collects messages are left silent here as this form of data collection technology is readily available in today's marketplace and computing systems, where ultra-mundane methods such as sending a self-contained e-mail to some e-mail address might even suffice as a methodology of counting (horribly inefficient but indicative of the vast arrays of methods to have a master identity database keep track of hundreds if not millions of copies of "itself").

In view of the wide variety of implementations and applications to which the principles of this technology can be put, it should be apparent that the detailed embodiments are illustrative only and in no way limit the scope of my invention. I claim as my invention all such embodiments as come within the scope and spirit of the following claims and equivalents thereto.

CLAIMS

1. A multi-computer system including a network for embedding and reading a watermark, the system comprising:

a first digital electrical computer system comprising a first digital electrical computer electrically connected to a first input device, to a first output device, and to a first memory storing a plurality of creator identifiers and creator contact data corresponding to each of the creator identifiers;

a second digital electrical computer system comprising a second digital electrical computer electrically connected to a second input device and to a second output device, the second digital electrical computer being programmed to embed a watermark in a digital photographic image, the watermark including one of the plurality of creator identifiers;

a third digital electrical computer system comprising a third digital electrical computer electrically connected to a third input device and to a third output device, the third digital electrical computer being programmed to read the watermark in the digital photographic image to reveal the one of the plurality of creator identifiers;

a network for communicating the revealed one of the plurality of creator identifiers to the first digital electrical computer to obtain the creator contact data corresponding to the one of the plurality of creator identifiers form the memory.

2. The system of claim 1, wherein:

the second digital electrical computer is programmed to automatically detect for a watermark when an image is first examined by the second digital electrical computer.

3. The system of claim 1, wherein:

the second digital electrical computer is programmed to selectably detect for a watermark when an image is examined by the second digital electrical computer.

4. The system of claim 1, wherein:

the network includes the Internet;

the watermark includes information identifying a World Wide Web site; and

wherein

the third digital electrical computer system is programmed to load a World Wide Web browser and connect to the World Wide Web site in response to the revealed one

of the plurality of creator identifiers.

5. The system of claim 1, wherein:

the watermark includes extended data including at least one member from the group consisting of an organization identifier, a transaction identifier, and an item identifier.

6. In a television set, audio stereo system, computing system, or other system capable of receiving, storing, transmitting, or processing an audio or visual signal, an improvement comprising:

means for adding to an initial audio or visual signal an additional digital information associated with the initial signal,

the additional information containing, in whole or in part, an identification number which is uniquely assigned to the individual initial signal or copies thereof,

wherein the identification number represents an address to a location in a remote database wherein a prestored set of information associated with the initial signal is contained, and

means coupled to the system containing a processor responsive to basic operating instructions for enabling the reading of the additional information and enabling the realtime retrieving from the remote database a subset of the prestored information via a communications link with the remote site.

7. The system of claim 6, wherein the prestored set of information comprises identification and description information about the signal or an object contained in or represented by the signal.

8. The system of claim 6, wherein the prestored set of information comprises a set of machine instructions executable to perform a task relative to the initial signal or about an object contained in the initial signal.

9. The system of claim 6, wherein the prestored set of information comprises a menu of options or commands.

10. The system of claim 1, including means for steganographically embedding the additional information in the initial signal.

11. The system of claim 1, wherein there also exists a second number as part of the additional information, wherein the second number is the address or identifier of the database itself to which the first identification points inside, in the context of

multiple databases serving a multitude of audio and visual signals.

12. The system of claim 4, first and second numbers steganographically embedded in the initial signal.

13. The system of claim 1, wherein there is additionally an instrument or receiving station which monitors a multiple number of audio/visual public channels searching and recording the identification numbers and reporting sitings.

14. In an image processing method that includes steganographically decoding an input two-dimensional image to extract a multi-bit code therein, the image comprising a two-dimensional array of pixels, an improvement comprising:

transforming the image into the spatial frequency domain;

pattern matching the transformed image so spatial frequencies obtained by said transforming step coincide with reference spatial frequencies, to thereby effect registration of the transformed image;

inverse-transforming the transformed image to yield a registered image;

identifying, in the registered image, a plurality of regions that encode a first control bit, said regions being distributed through the registered image in a regular array;

performing a statistical analysis over at least said plurality of regions to determine whether the first control bit has first or second values;

if said control bit has the first value, performing a first decoding process on the image to extract the code therefrom; and

if said control bit has the second value, performing a second decoding process on the image to extract the code therefrom, the second decoding process being different than the first.

FIG. 1

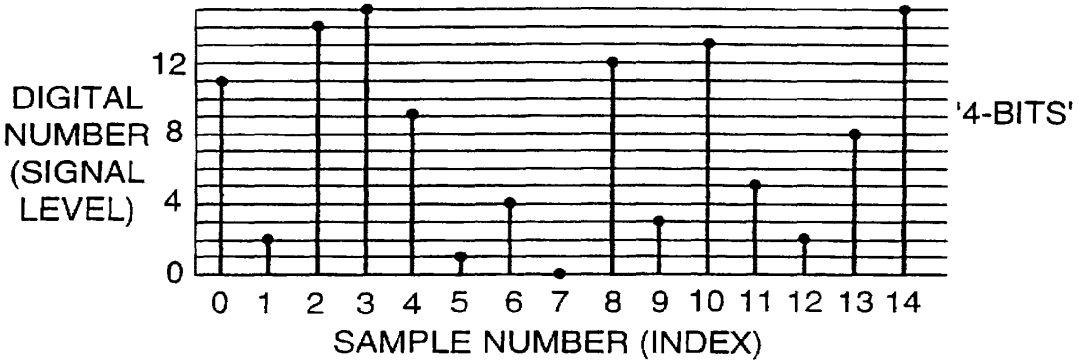

DIGITAL NUMBER (SIGNAL LEVEL)
12
8
4
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
SAMPLE NUMBER (INDEX)
'4-BITS'

FIG. 4

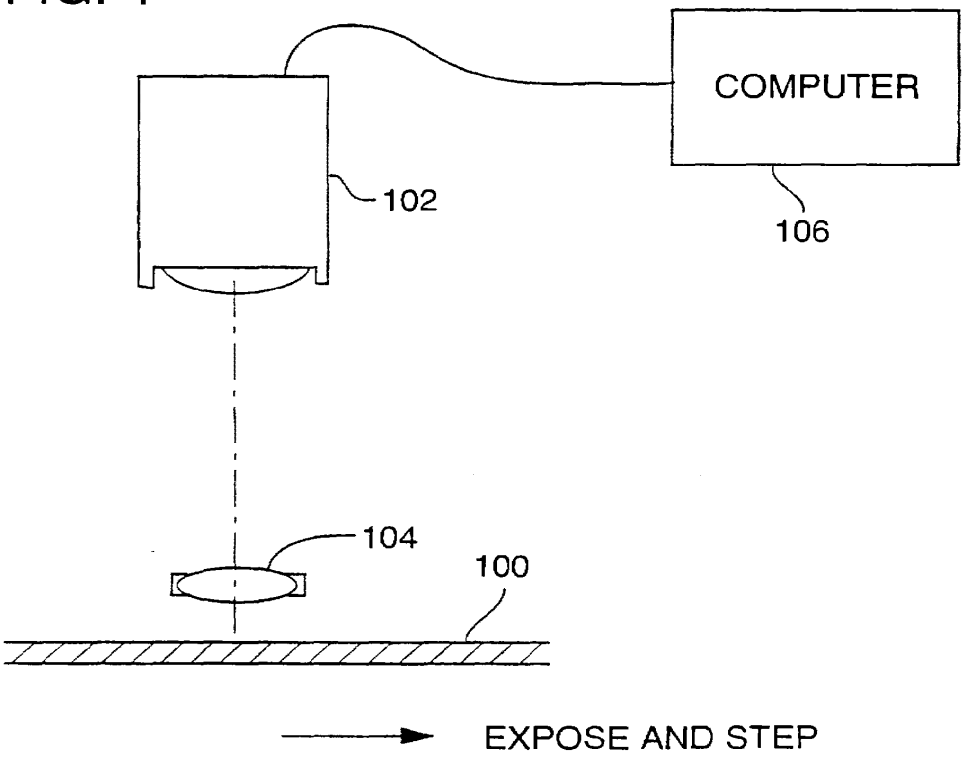

COMPUTER
102
106
104
100
EXPOSE AND STEP

FIG. 2

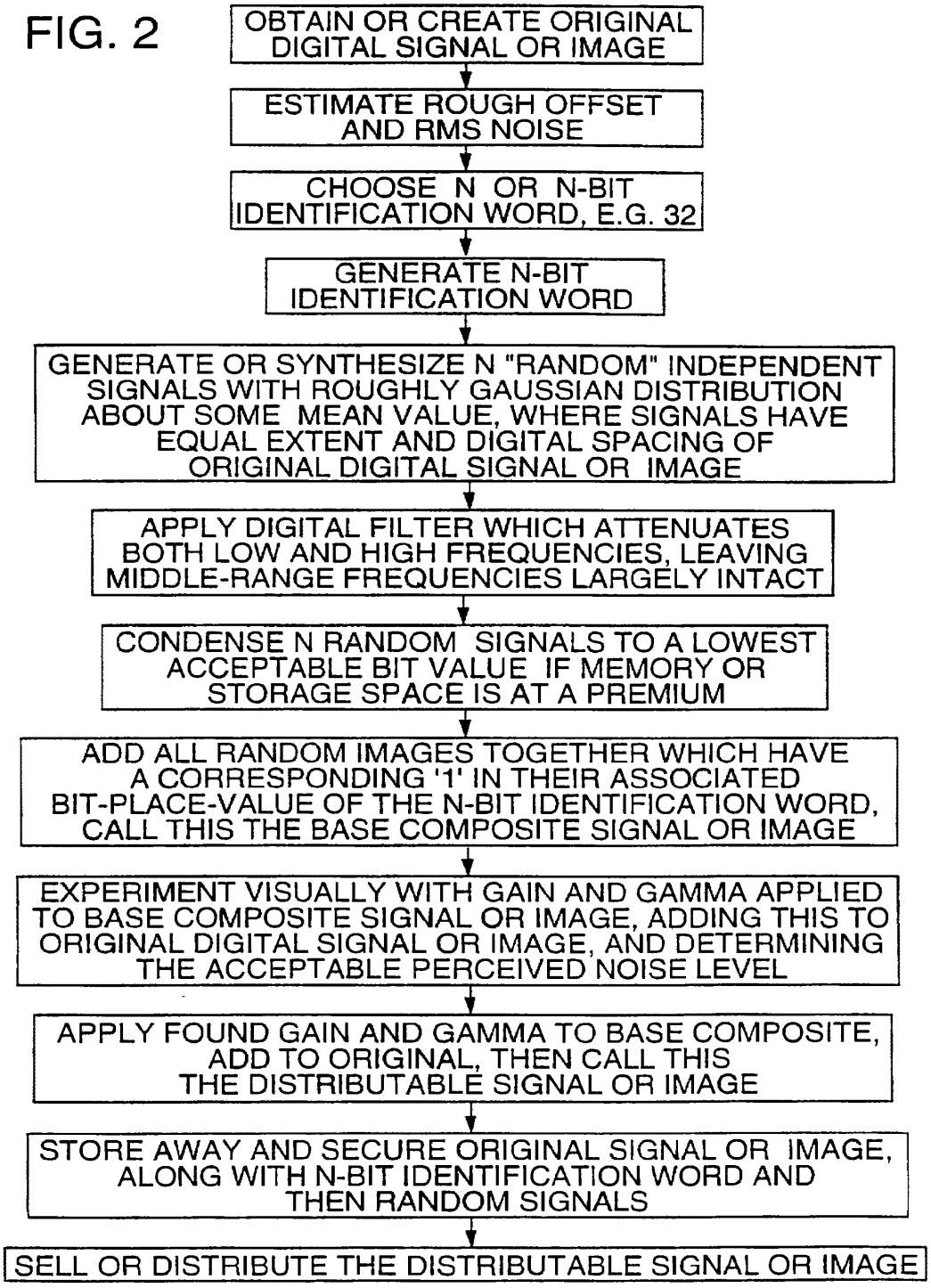

OBTAIN OR CREATE ORIGINAL DIGITAL SIGNAL OR IMAGE
ESTIMATE ROUGH OFFSET AND RMS NOISE
CHOOSE N OR N-BIT IDENTIFICATION WORD, E.G. 32
GENERATE N-BIT IDENTIFICATION WORD
GENERATE OR SYNTHESIZE N "RANDOM" INDEPENDENT SIGNALS WITH ROUGHLY GAUSSIAN DISTRIBUTION ABOUT SOME MEAN VALUE, WHERE SIGNALS HAVE EQUAL EXTENT AND DIGITAL SPACING OF ORIGINAL DIGITAL SIGNAL OR IMAGE
APPLY DIGITAL FILTER WHICH ATTENUATES BOTH LOW AND HIGH FREQUENCIES, LEAVING MIDDLE-RANGE FREQUENCIES LARGELY INTACT
CONDENSE N RANDOM SIGNALS TO A LOWEST ACCEPTABLE BIT VALUE IF MEMORY OR STORAGE SPACE IS AT A PREMIUM
ADD ALL RANDOM IMAGES TOGETHER WHICH HAVE A CORRESPONDING '1' IN THEIR ASSOCIATED BIT-PLACE-VALUE OF THE N-BIT IDENTIFICATION WORD, CALL THIS THE BASE COMPOSITE SIGNAL OR IMAGE
EXPERIMENT VISUALLY WITH GAIN AND GAMMA APPLIED TO BASE COMPOSITE SIGNAL OR IMAGE, ADDING THIS TO ORIGINAL DIGITAL SIGNAL OR IMAGE, AND DETERMINING THE ACCEPTABLE PERCEIVED NOISE LEVEL
APPLY FOUND GAIN AND GAMMA TO BASE COMPOSITE, ADD TO ORIGINAL, THEN CALL THIS THE DISTRIBUTABLE SIGNAL OR IMAGE
STORE AWAY AND SECURE ORIGINAL SIGNAL OR IMAGE, ALONG WITH N-BIT IDENTIFICATION WORD AND THEN RANDOM SIGNALS
SELL OR DISTRIBUTE THE DISTRIBUTABLE SIGNAL OR IMAGE

FIG. 3

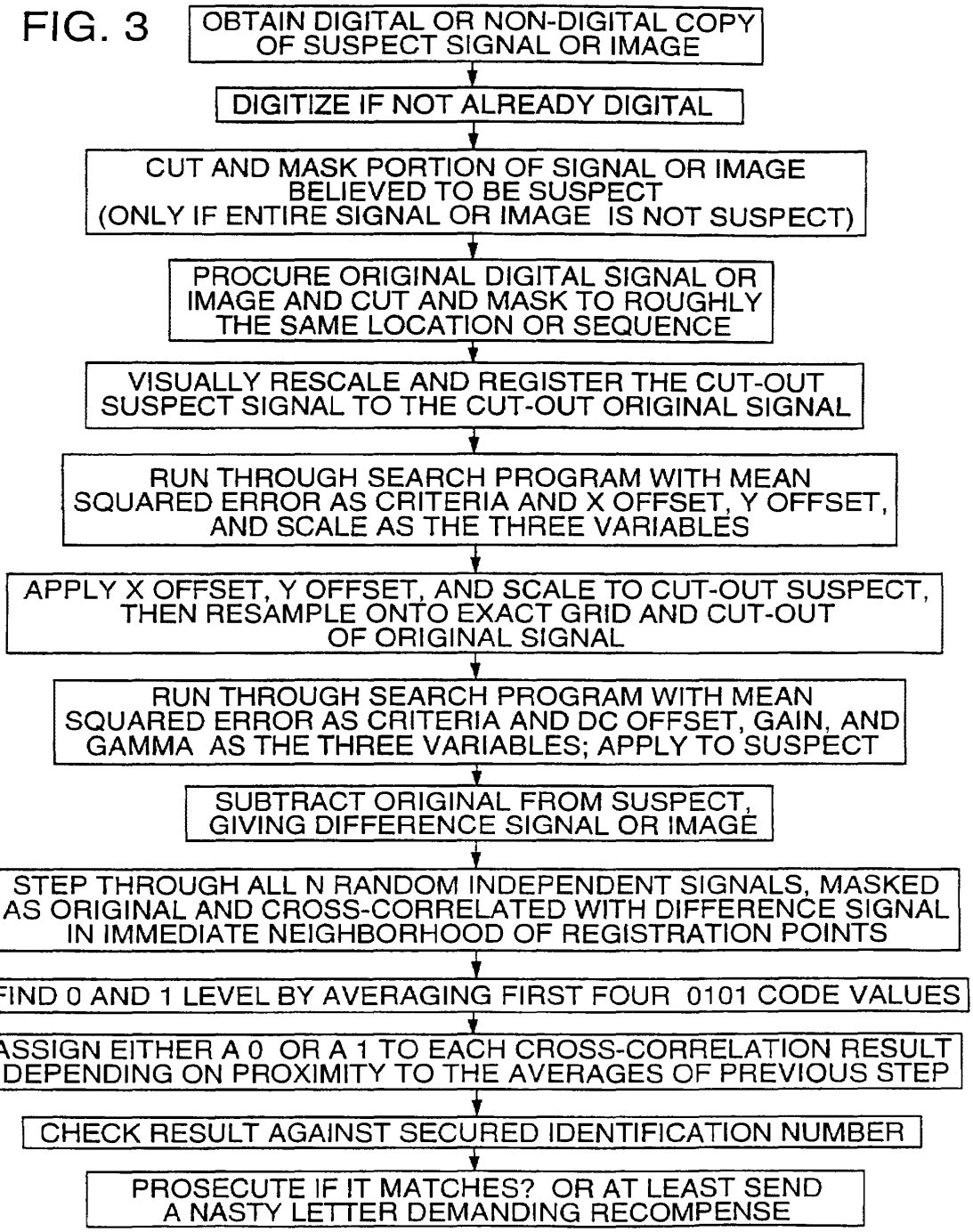

OBTAIN DIGITAL OR NON-DIGITAL COPY OF SUSPECT SIGNAL OR IMAGE
DIGITIZE IF NOT ALREADY DIGITAL
CUT AND MASK PORTION OF SIGNAL OR IMAGE BELIEVED TO BE SUSPECT (ONLY IF ENTIRE SIGNAL OR IMAGE IS NOT SUSPECT)
PROCURE ORIGINAL DIGITAL SIGNAL OR IMAGE AND CUT AND MASK TO ROUGHLY THE SAME LOCATION OR SEQUENCE
VISUALLY RESCALE AND REGISTER THE CUT-OUT SUSPECT SIGNAL TO THE CUT-OUT ORIGINAL SIGNAL
RUN THROUGH SEARCH PROGRAM WITH MEAN SQUARED ERROR AS CRITERIA AND X OFFSET, Y OFFSET, AND SCALE AS THE THREE VARIABLES
APPLY X OFFSET, Y OFFSET, AND SCALE TO CUT-OUT SUSPECT, THEN RESAMPLE ONTO EXACT GRID AND CUT-OUT OF ORIGINAL SIGNAL
RUN THROUGH SEARCH PROGRAM WITH MEAN SQUARED ERROR AS CRITERIA AND DC OFFSET, GAIN, AND GAMMA AS THE THREE VARIABLES; APPLY TO SUSPECT
SUBTRACT ORIGINAL FROM SUSPECT, GIVING DIFFERENCE SIGNAL OR IMAGE
STEP THROUGH ALL N RANDOM INDEPENDENT SIGNALS, MASKED AS ORIGINAL AND CROSS-CORRELATED WITH DIFFERENCE SIGNAL IN IMMEDIATE NEIGHBORHOOD OF REGISTRATION POINTS
FIND 0 AND 1 LEVEL BY AVERAGING FIRST FOUR 0101 CODE VALUES
ASSIGN EITHER A 0 OR A 1 TO EACH CROSS-CORRELATION RESULT DEPENDING ON PROXIMITY TO THE AVERAGES OF PREVIOUS STEP
CHECK RESULT AGAINST SECURED IDENTIFICATION NUMBER
PROSECUTE IF IT MATCHES? OR AT LEAST SEND A NASTY LETTER DEMANDING RECOMPENSE

FIG. 5

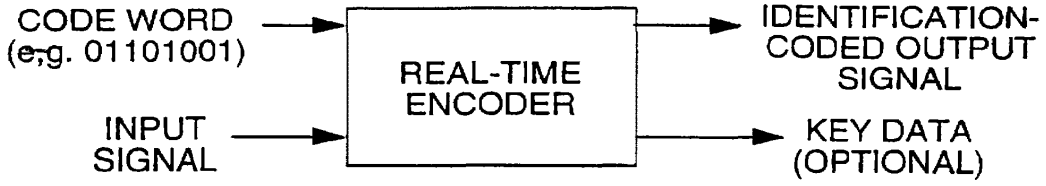
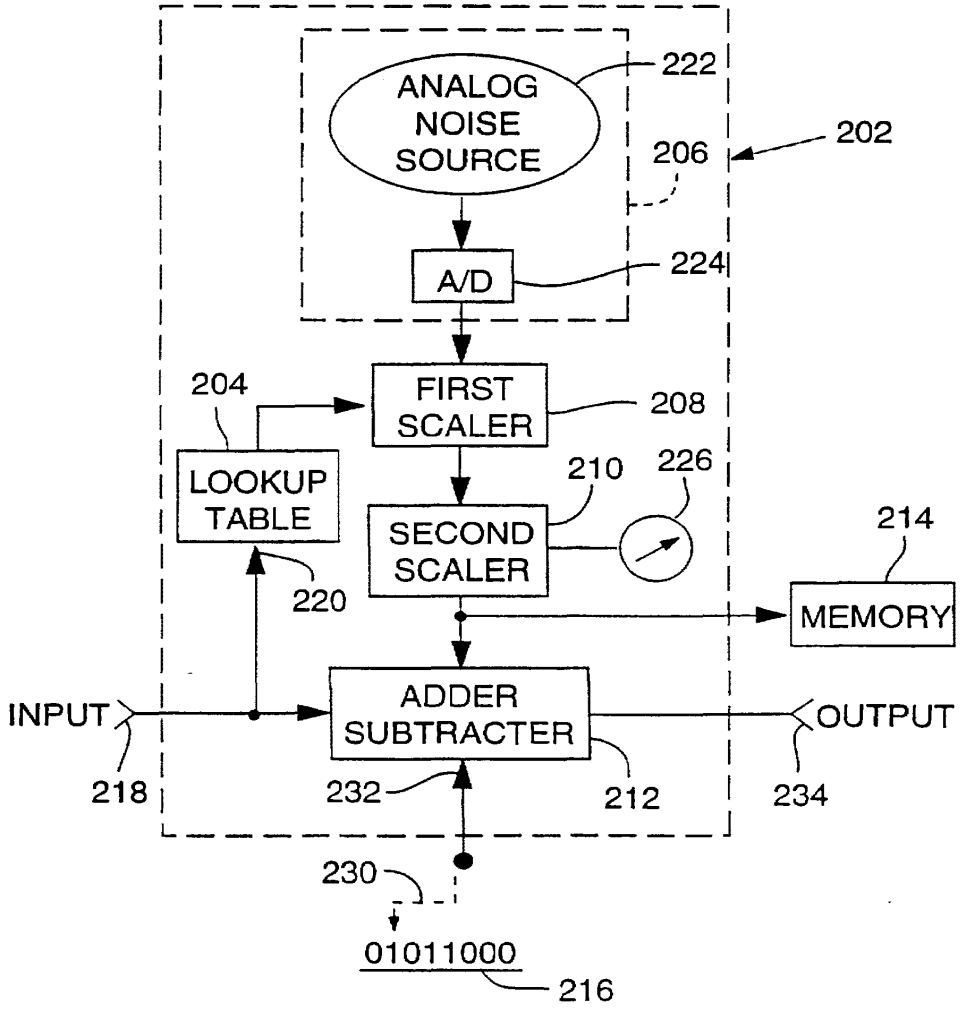

CODE WORD
(e.g. 01101001)
INPUT
SIGNAL
REAL-TIME
ENCODER
IDENTIFICATION-
CODED OUTPUT
SIGNAL
KEY DATA
(OPTIONAL)

FIG. 6


ANALOG
NOISE
SOURCE
222
202
206
A/D
224
204
FIRST
SCALER
208
LOOKUP
TABLE
210
226
SECOND
SCALER
214
220
MEMORY
INPUT
ADDER
SUBTRACTER
OUTPUT
218
232
212
234
230
01011000
216

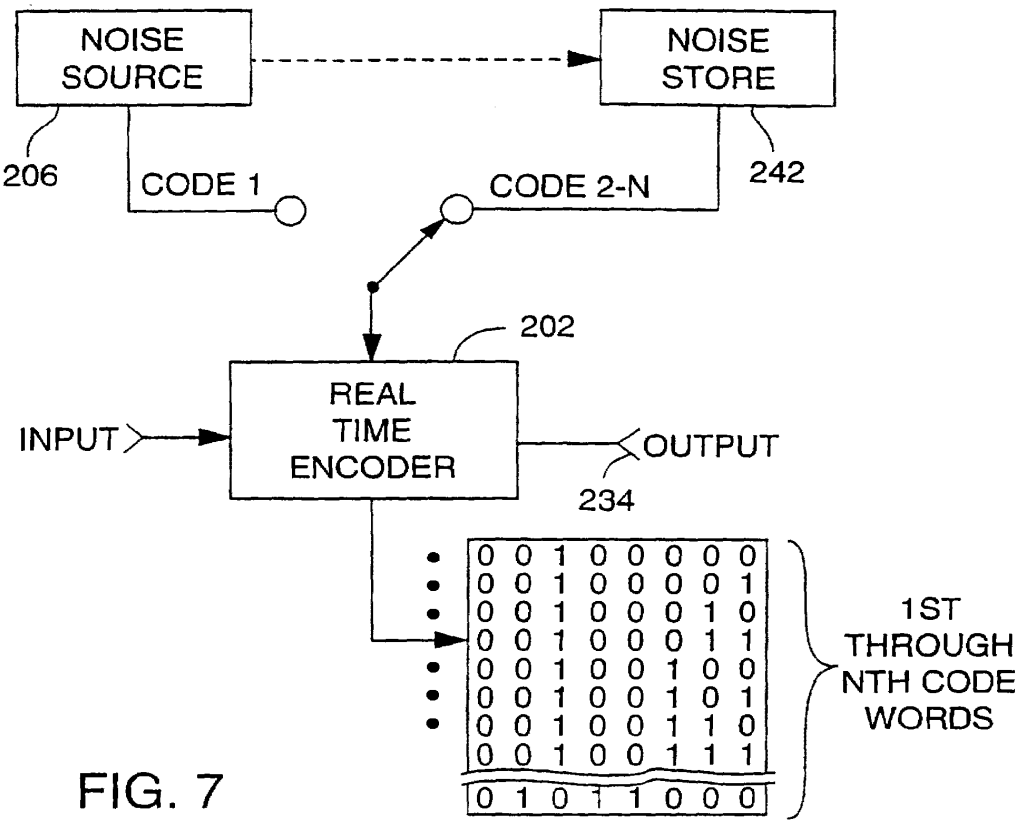

NOISE SOURCE
NOISE STORE
206
CODE 1
CODE 2-N
242
202
REAL TIME ENCODER
INPUT
OUTPUT
234
0 0 1 0 0 0 0 0
0 0 1 0 0 0 0 1
0 0 1 0 0 0 1 0
0 0 1 0 0 0 1 1
0 0 1 0 0 1 0 0
0 0 1 0 0 1 0 1
0 0 1 0 0 1 1 0
0 0 1 0 0 1 1 1
0 1 0 1 1 0 0 0
1ST THROUGH NTH CODE WORDS

FIG. 7

FIG. 8

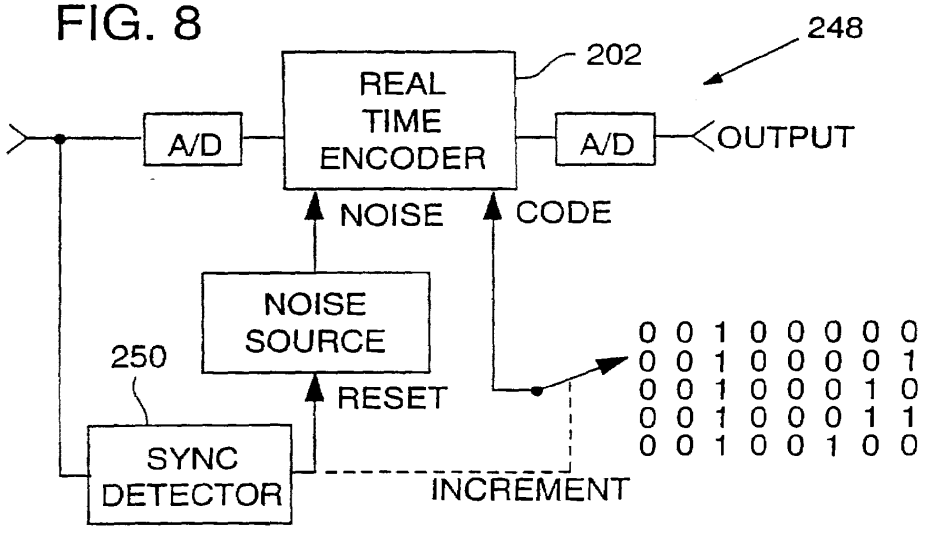

248
202
REAL TIME ENCODER
A/D
A/D
OUTPUT
NOISE
CODE
NOISE SOURCE
250
RESET
SYNC DETECTOR
INCREMENT
0 0 1 0 0 0 0 0
0 0 1 0 0 0 0 1
0 0 1 0 0 0 1 0
0 0 1 0 0 0 1 1
0 0 1 0 0 1 0 0

FIG. 9A

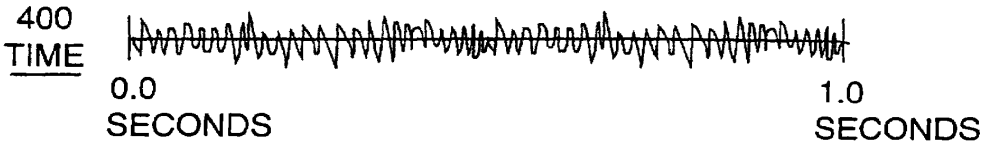


FIG. 9B

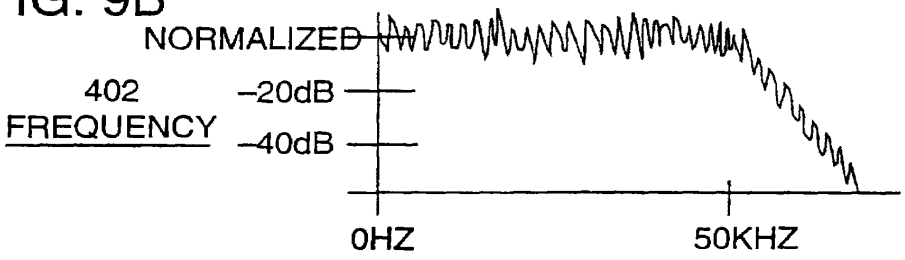


FIG. 9C

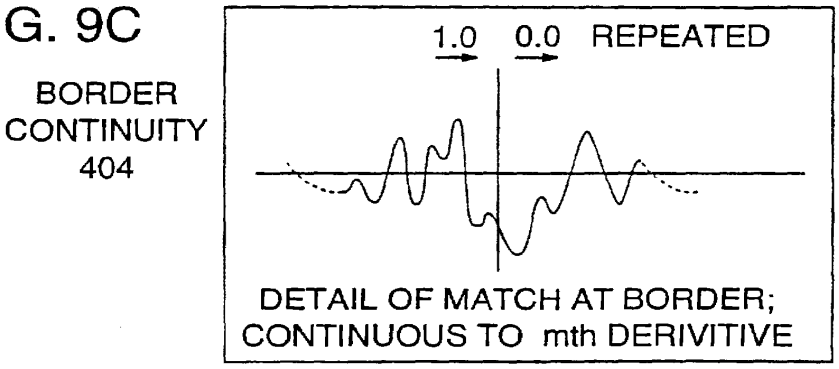


FIG. 10

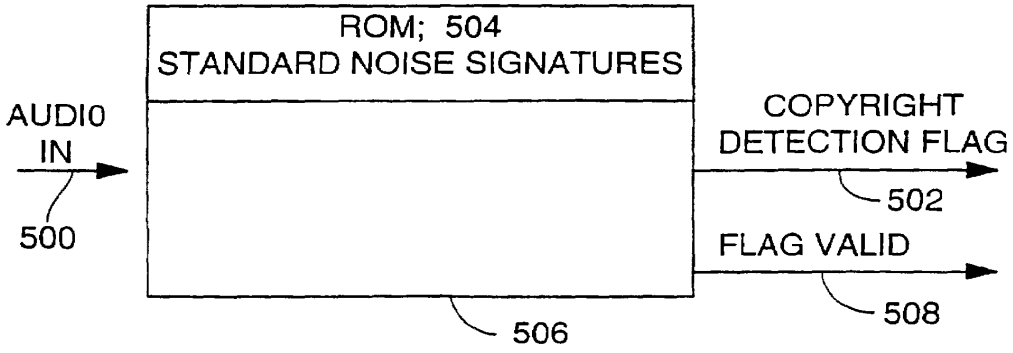


FIG. 11

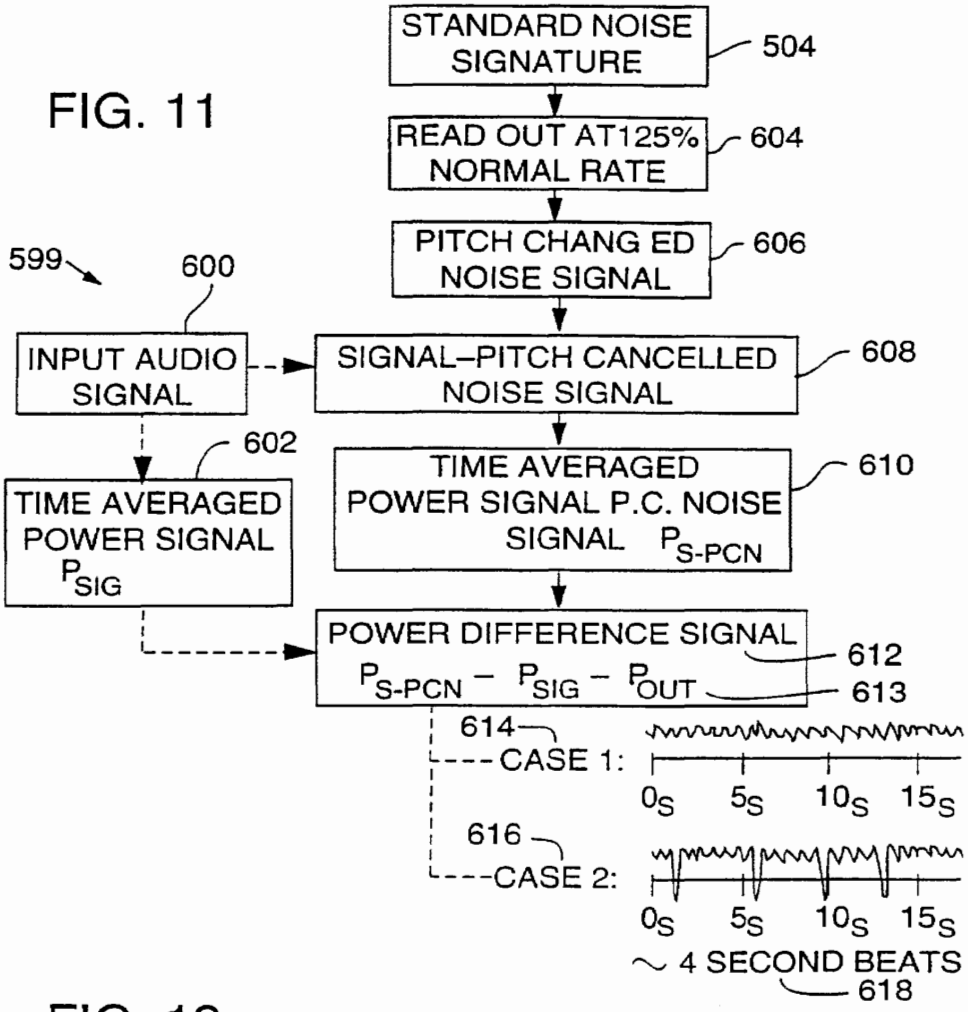

STANDARD NOISE SIGNATURE
504
READ OUT AT125% NORMAL RATE
604
PITCH CHANG ED NOISE SIGNAL
606
599
600
INPUT AUDIO SIGNAL
SIGNAL–PITCH CANCELLED NOISE SIGNAL
608
602
TIME AVERAGED POWER SIGNAL P_{SIG}
TIME AVERAGED POWER SIGNAL P.C. NOISE SIGNAL P_{S-PCN}
610
POWER DIFFERENCE SIGNAL
$P_{S-PCN} - P_{SIG} - P_{OUT}$
612
613
614
CASE 1:
0S 5S 10S 15S
616
CASE 2:
0S 5S 10S 15S
4 SECOND BEATS
618

FIG. 12

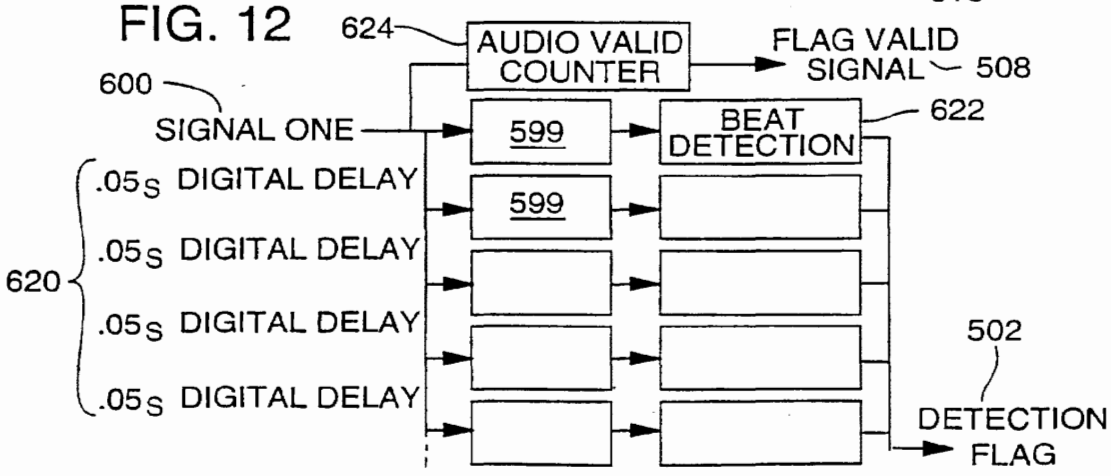
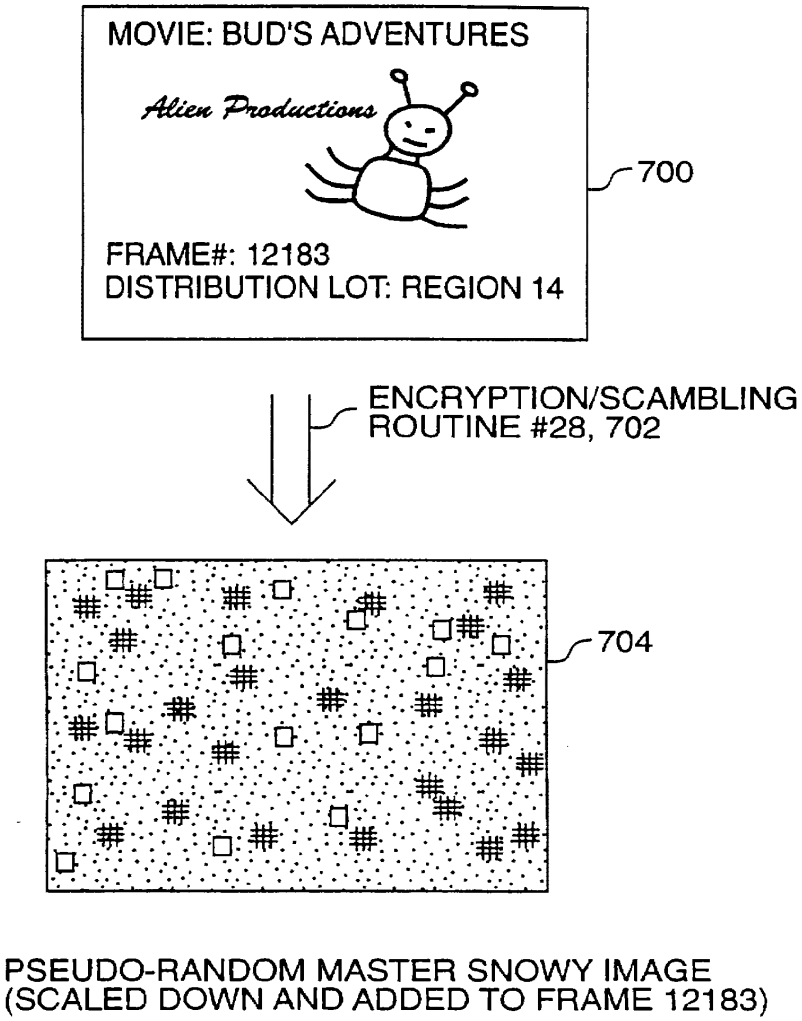
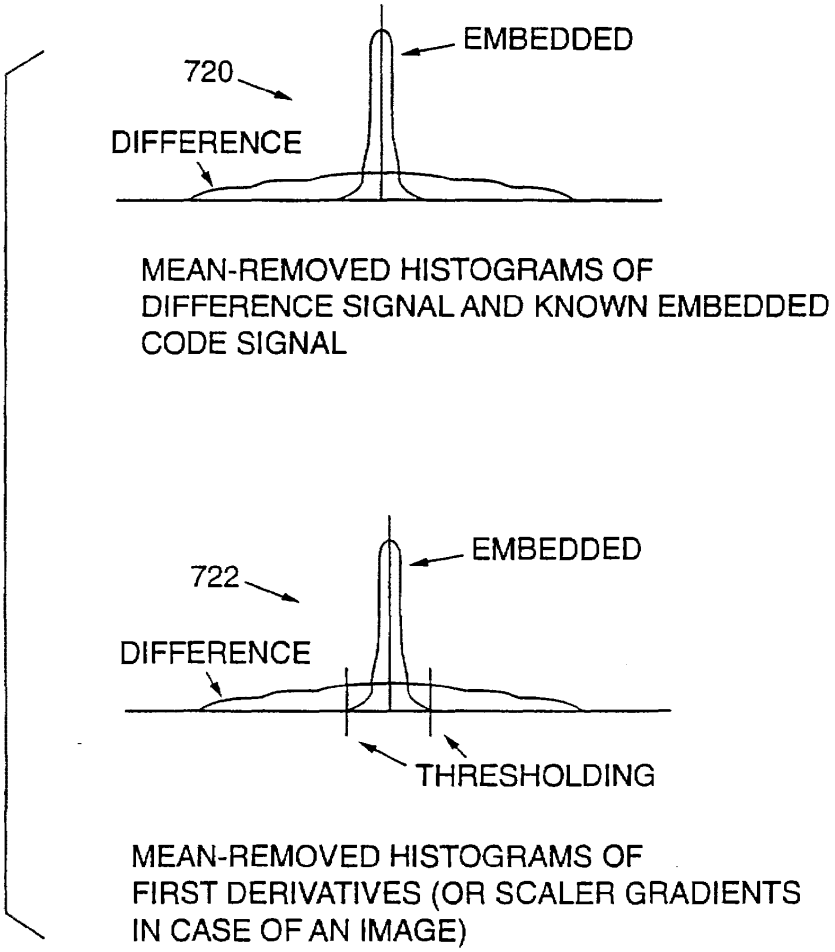

624
AUDIO VALID COUNTER
FLAG VALID SIGNAL
508
600
SIGNAL ONE
599
BEAT DETECTION
622
.05S DIGITAL DELAY
599
.05S DIGITAL DELAY
620
.05S DIGITAL DELAY
.05S DIGITAL DELAY
502
DETECTION FLAG

FIG. 13



PSEUDO-RANDOM MASTER SNOWY IMAGE
(SCALED DOWN AND ADDED TO FRAME 12183)

FIG. 14


EMBEDDED
720
DIFFERENCE
MEAN-REMOVED HISTOGRAMS OF DIFFERENCE SIGNAL AND KNOWN EMBEDDED CODE SIGNAL
EMBEDDED
722
DIFFERENCE
THRESHOLDING
MEAN-REMOVED HISTOGRAMS OF FIRST DERIVATIVES (OR SCALER GRADIENTS IN CASE OF AN IMAGE)

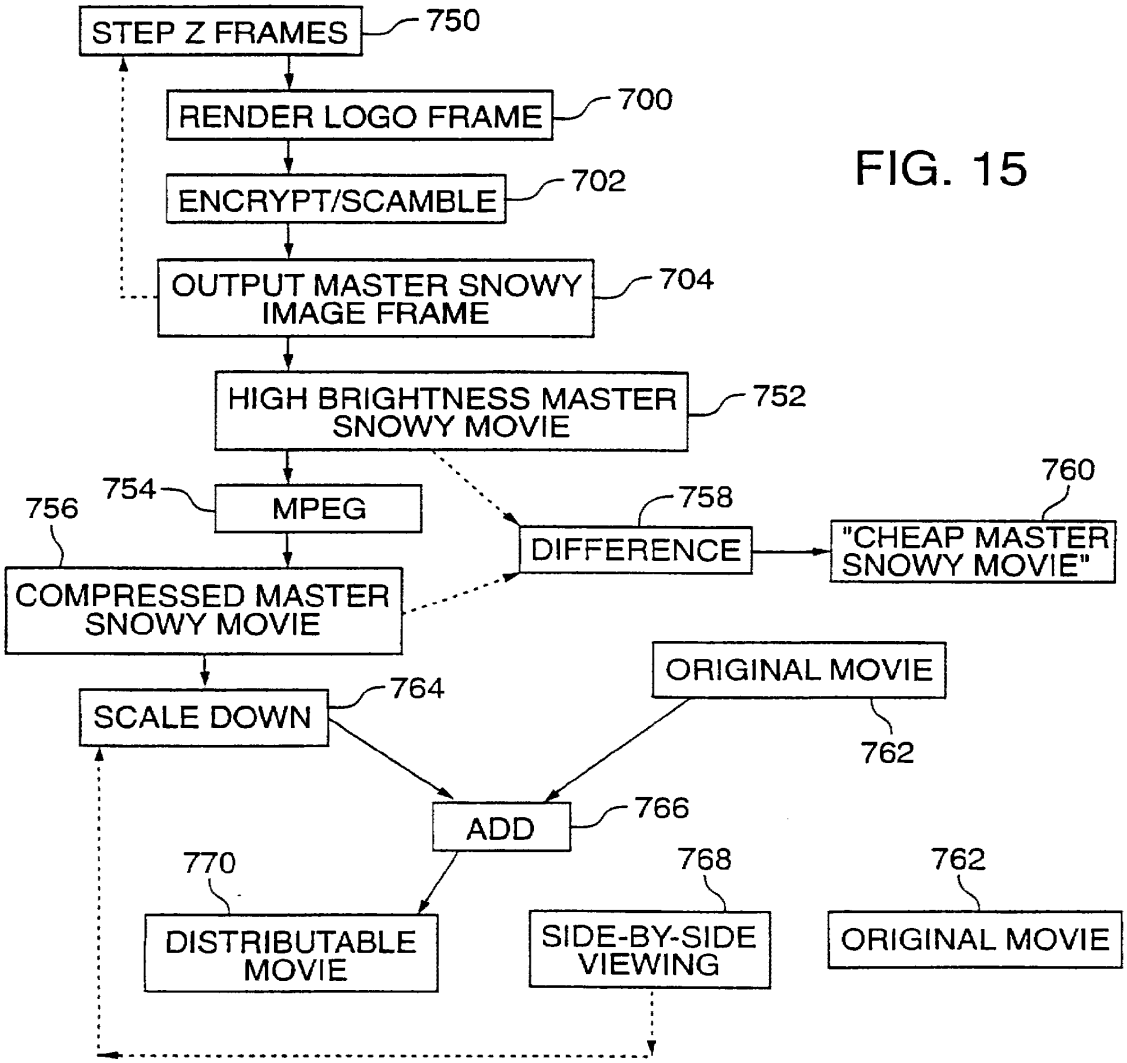

STEP Z FRAMES
750
RENDER LOGO FRAME
700
ENCRYPT/SCAMBLE
702
OUTPUT MASTER SNOWY IMAGE FRAME
704
HIGH BRIGHTNESS MASTER SNOWY MOVIE
752
754
MPEG
756
COMPRESSED MASTER SNOWY MOVIE
758
DIFFERENCE
760
"CHEAP MASTER SNOWY MOVIE"
ORIGINAL MOVIE
762
764
SCALE DOWN
766
ADD
770
DISTRIBUTABLE MOVIE
768
SIDE-BY-SIDE VIEWING
762
ORIGINAL MOVIE

FIG. 15

FIG. 16

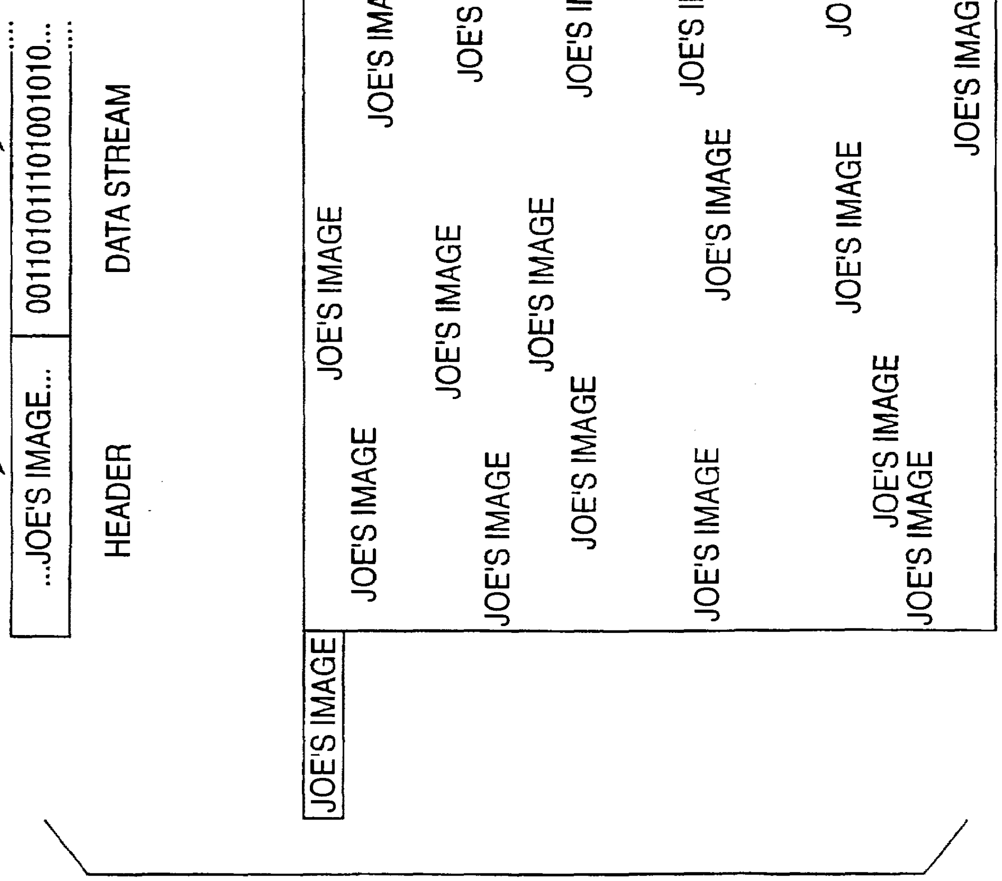
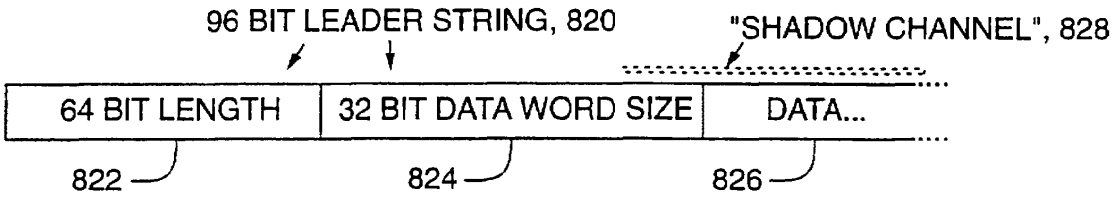

800
802
...JOE'S IMAGE...
0011010111010001010...
HEADER
DATA STREAM
JOE'S IMAGE

FIG. 17



UNIVERSAL EMPIRICAL DATA FORMAT

FIG. 18

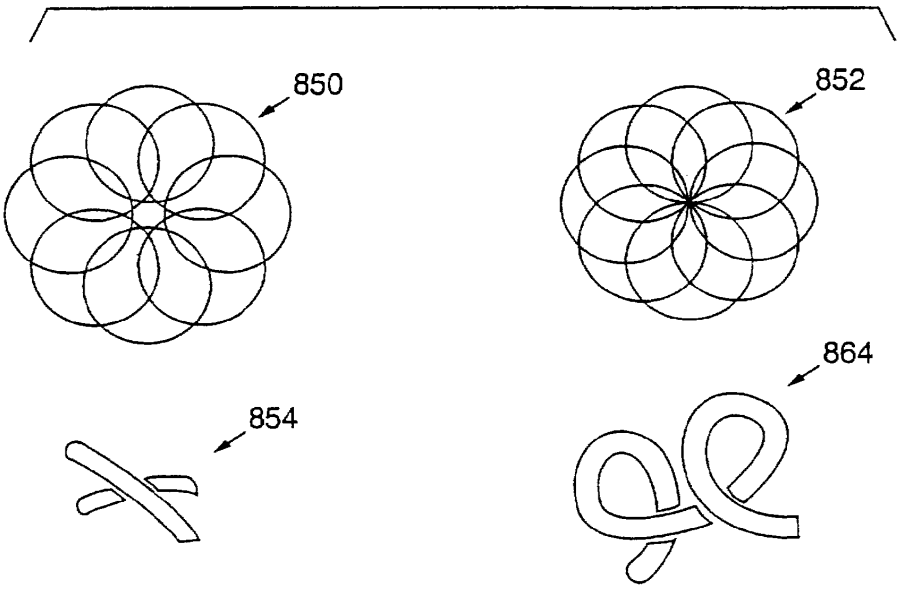
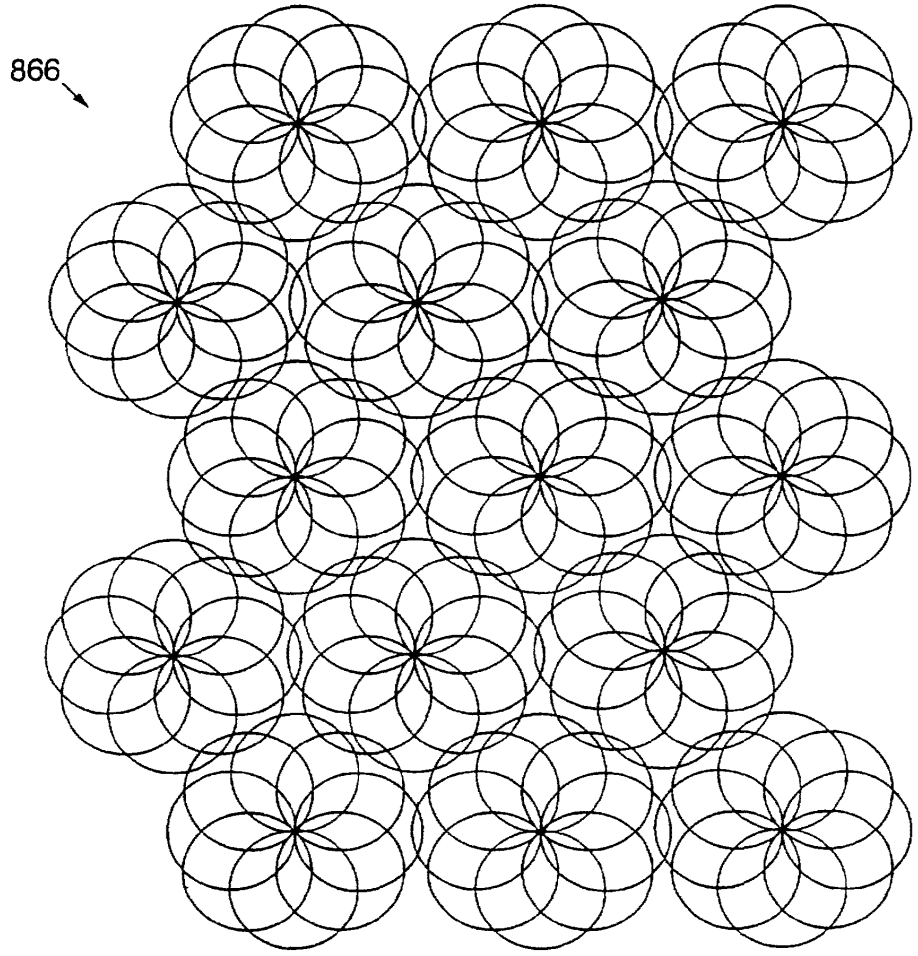


FIG. 19



QUEST FOR MOSAICED KNOT PATTERNS WHICH "COVER" AND ARE COEXTENSIVE WITH ORIGINAL IMAGE;
ALL ELEMENTAL KNOT PATTERNS CAN CONVEY THE SAME INFORMATION, SUCH AS A SIGNATURE, OR EACH CAN CONVEY A NEW MESSAGE IN A STEGANOGRAPHIC SENSE

FIG. 20

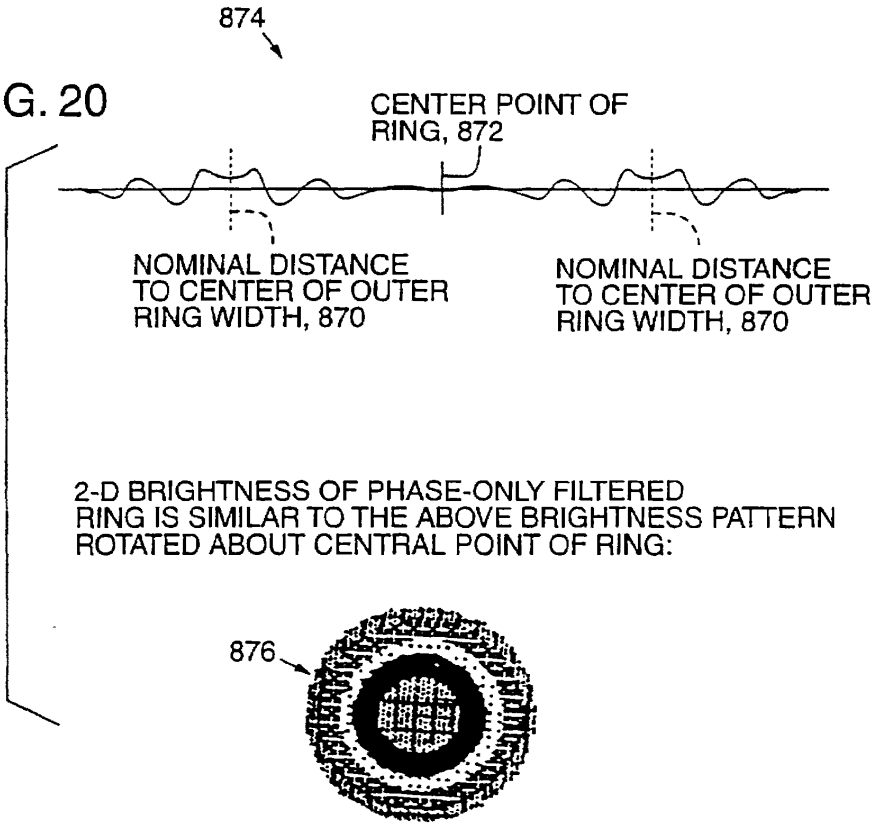
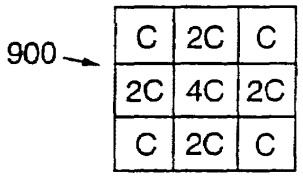

874
CENTER POINT OF RING, 872
NOMINAL DISTANCE TO CENTER OF OUTER RING WIDTH, 870
NOMINAL DISTANCE TO CENTER OF OUTER RING WIDTH, 870
2-D BRIGHTNESS OF PHASE-ONLY FILTERED RING IS SIMILAR TO THE ABOVE BRIGHTNESS PATTERN ROTATED ABOUT CENTRAL POINT OF RING:
876

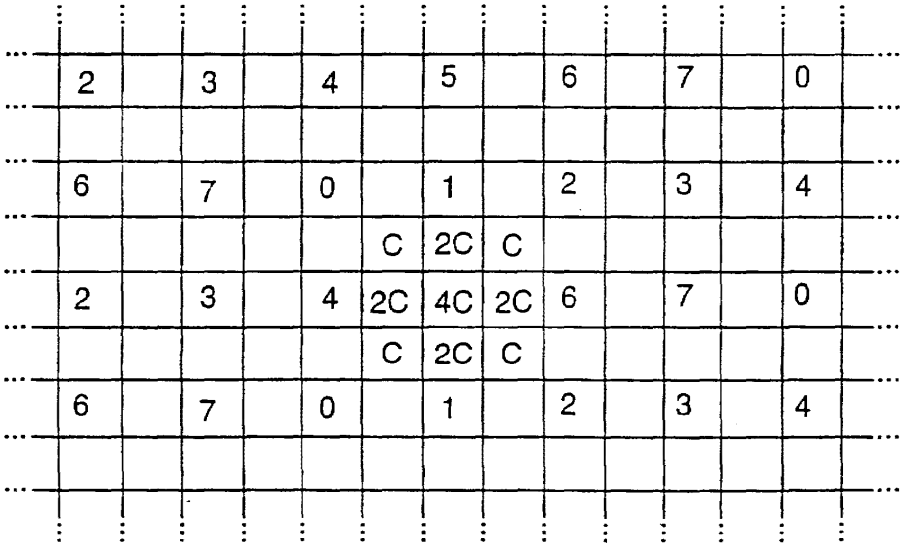
FIG. 21A



WHERE C = 1/16

ELEMENTARY BUMP
(DEFINED GROUPING OF PIXELS WITH WEIGHT VALUES)

FIG. 21B



EXAMPLE OF HOW MANY ELEMENTARY BUMPS, 900, WOULD BE ASSIGNED LOCATIONS IN AN IMAGE, AND THOSE LOCATIONS WOULD BE ASSOCIATED WITH A CORRESPONDING BIT PLANE IN THE N-BIT WORD, HERE TAKEN AS N=8 WITH INDEXES OF 0-7. ONE LOCATION, ASSOCIATED WITH BIT PLANE "5", HAS THE OVERLAY OF THE BUMP PROFILE DEPICTED.

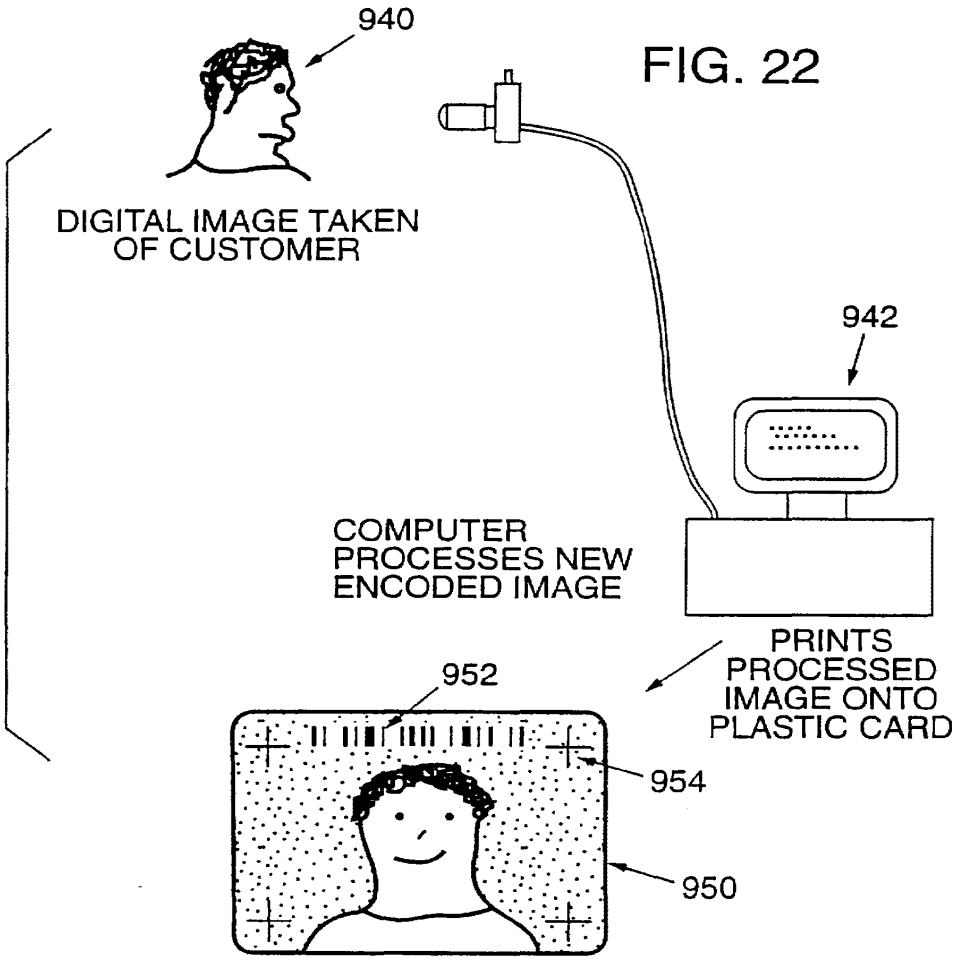
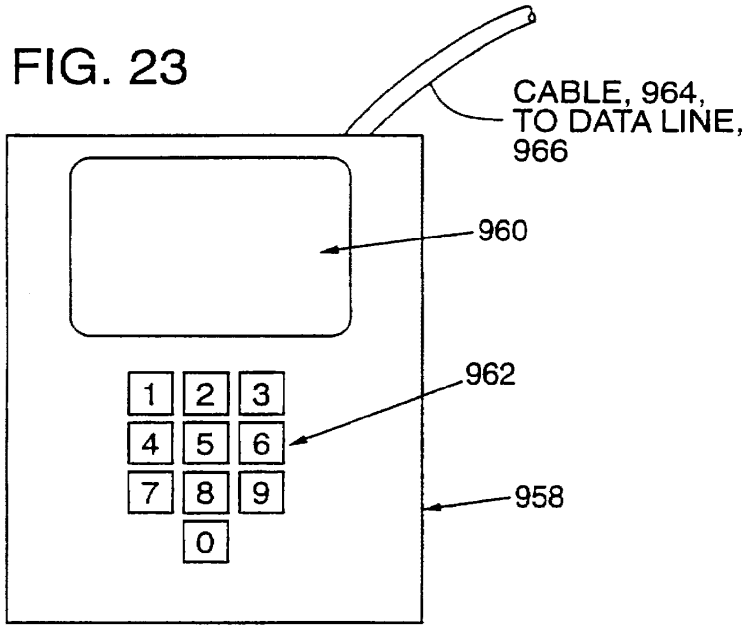

940
DIGITAL IMAGE TAKEN OF CUSTOMER
942
COMPUTER PROCESSES NEW ENCODED IMAGE
PRINTS PROCESSED IMAGE ONTO PLASTIC CARD
952
954
950

FIG. 22

FIG. 23



CONTAINS RUDIMENTARY OPTICAL SCANNER, MEMORY BUFFERS, COMMUNICATIONS DEVICES, AND MICROPROCESSOR

CONSUMER MERELY PLACES CARD INTO WINDOW AND CAN, AT THEIR PREARRANGED OPTION, EITHER TYPE IN A PERSONAL IDENTIFICATION NUMBER (PIN, FOR ADDED SECURITY) OR NOT. THE TRANSACTION IS APPROVED OR DISAPPROVED WITHIN SECONDS.

FIG. 24

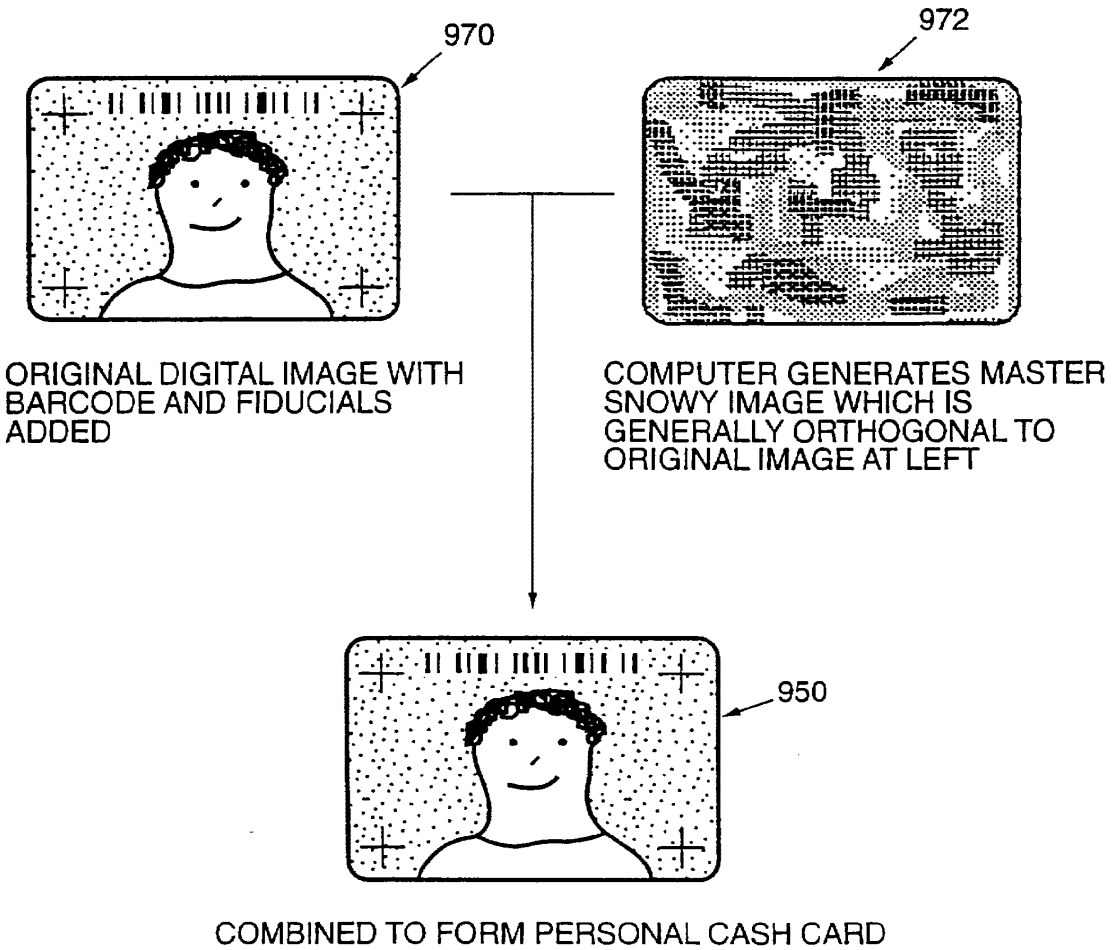

970
972
ORIGINAL DIGITAL IMAGE WITH BARCODE AND FIDUCIALS ADDED
COMPUTER GENERATES MASTER SNOWY IMAGE WHICH IS GENERALLY ORTHOGONAL TO ORIGINAL IMAGE AT LEFT
950
COMBINED TO FORM PERSONAL CASH CARD

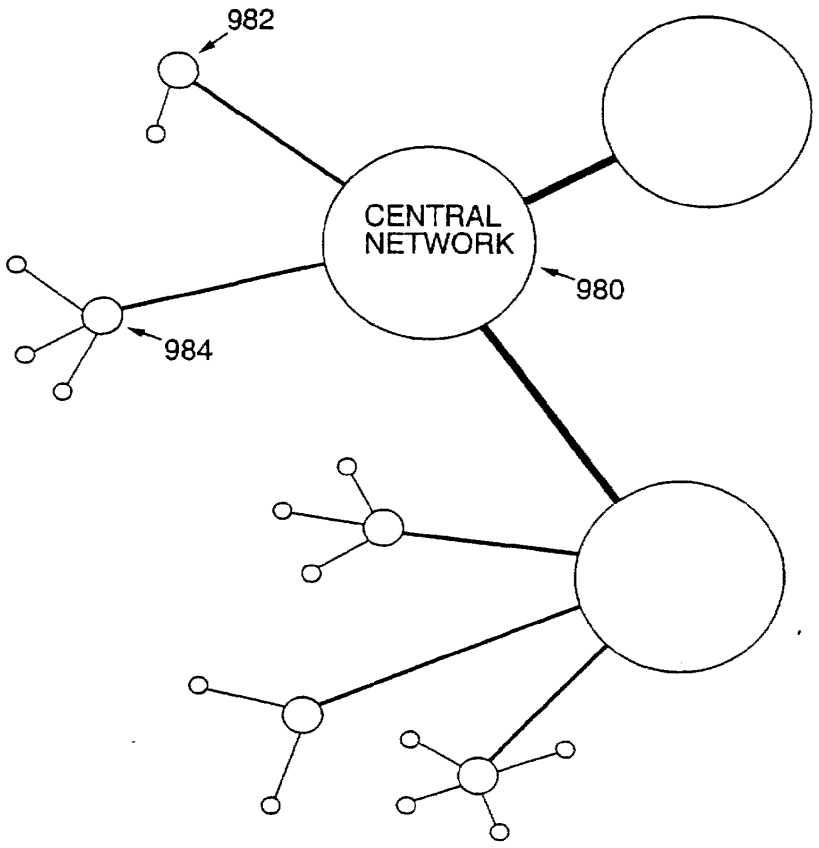
FIG. 25 TYPICAL TRANSACTION STEPS

1. READER SCANS IMAGE ON CARD, STORES IN MEMORY, EXTRACTS PERSON'S ID
2. OPTIONAL: USER KEYS IN PIN NUMBER
3. READER CALLS CENTRAL ACCOUNT DATA NETWORK, HANDSHAKES
4. READER SENDS ID, (PIN), MERCHANT INFORMATION, AND REQUESTED TRANSACTION AMOUNT TO CENTRAL NETWORK
5. CENTRAL NETWORK VERIFIES ID, PIN, MERCHANT INFO, AND ACCOUNT BALANCE
6. IF OK, CENTRAL NETWORK GENERATES TWENTY-FOUR SETS OF SIXTEEN DISTINCT RANDOM NUMBERS, WHERE THE RANDOM NUMBERS ARE INDEXES TO A SET OF 64K ORTHOGONAL SPATIAL PATTERNS
7. CENTRAL NETWORK TRANSMITS FIRST OK, AND THE SETS OF RANDOM NUMBERS
8. READER STEPS THROUGH THE TWENTY-FOUR SETS
   8A. READER ADDS TOGETHER SET OF ORTHOGONAL PATTERNS
   8B. READER PERFORMS DOT PRODUCT OF RESULTANT PATTERN AND CARD SCAN, STORES RESULT
9. READER TRANSMITS THE TWENTY-FOUR DOT PRODUCT RESULTS TO CENTRAL NETWORK
10. CENTRAL NETWORK CHECKS RESULTS AGAINST MASTER
11. CENTRAL NETWORK SENDS FINAL APPROVAL OR DENIAL
12. CENTRAL NETWORK DEBITS MERCHANT ACCOUNT, CREDITS CARD ACCOUNT

Ex.1002 / Page 352 of 1311
APPLE INC.

FIG. 26

THE NEGLIGIBLE-FRAUD CASH CARD SYSTEM



A BASIC FOUNDATION OF THE CASH CARD SYSTEM IS A 24-HOUR INFORMATION NETWORK, WHERE BOTH THE STATIONS WHICH CREATE THE PHYSICAL CASH CARDS, 950, AND THE POINT-OF-SALES, 984, ARE ALL HOOKED UP TO THE SAME NETWORK CONTINUOUSLY

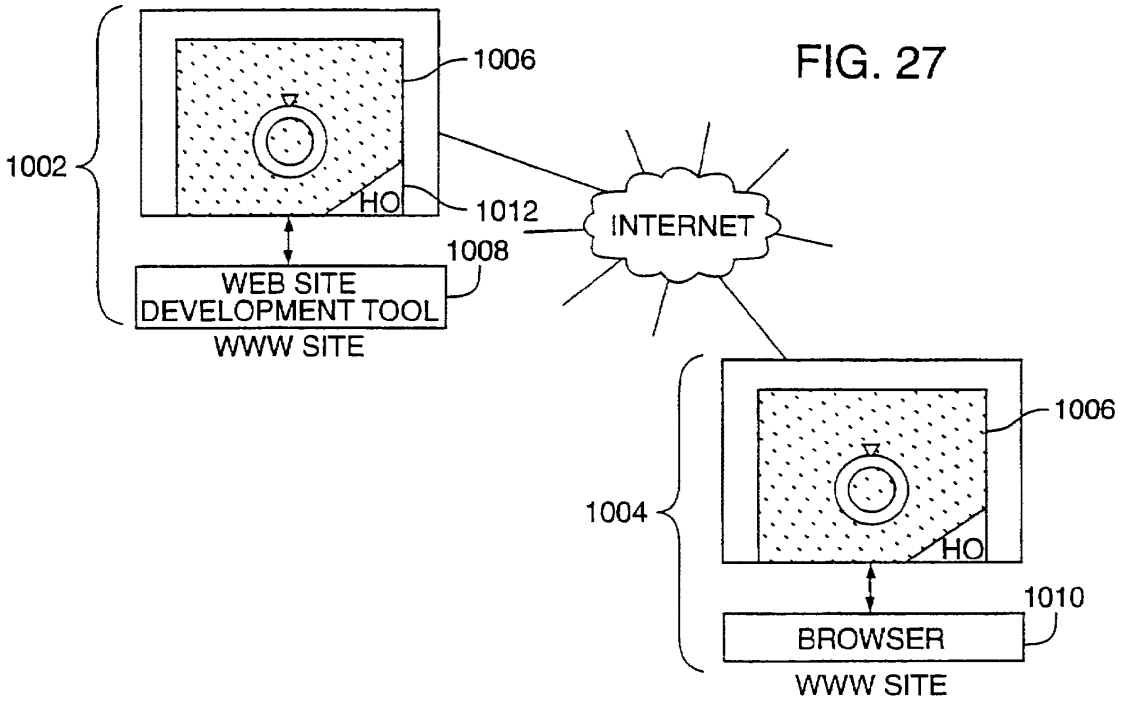

1002
1006
HO
1012
INTERNET
1008
WEB SITE
DEVELOPMENT TOOL
WWW SITE
1004
1006
HO
1010
BROWSER
WWW SITE

FIG. 27

FIG. 28

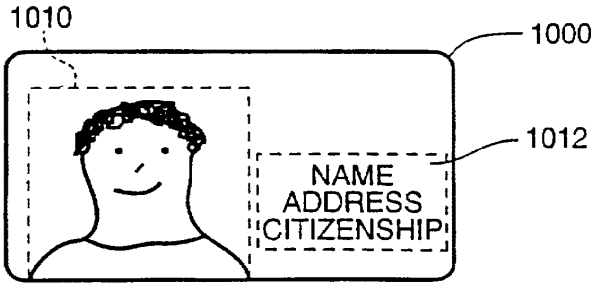

1010
1000
1012
NAME
ADDRESS
CITIZENSHIP

FIG. 27A

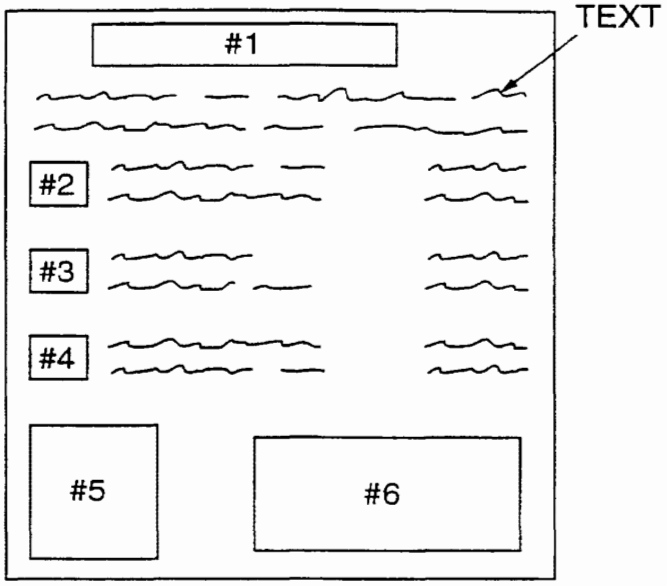


FIG. 27B

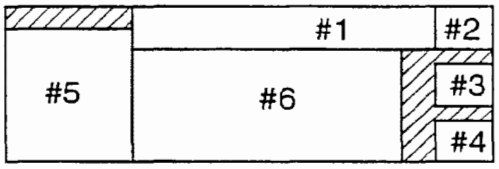
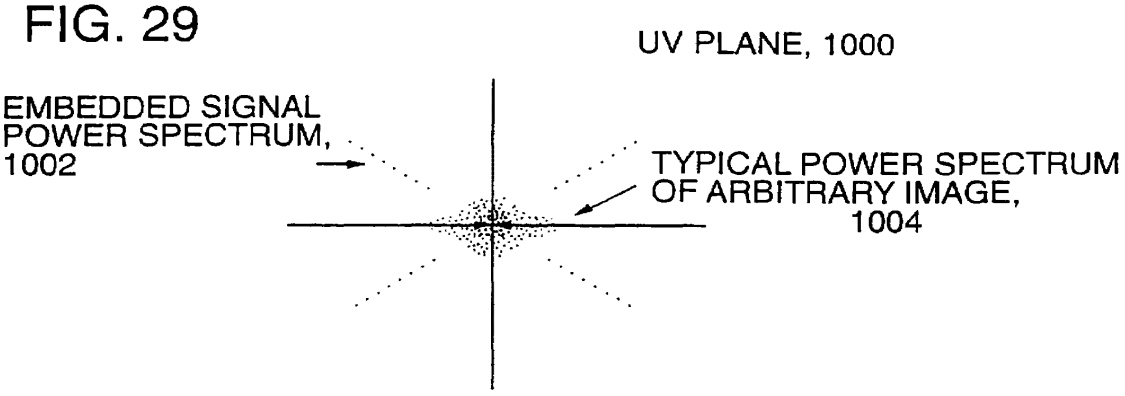
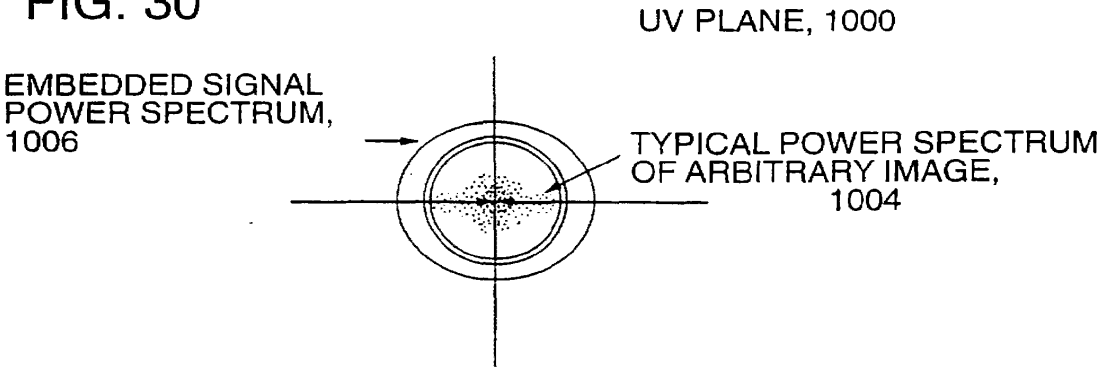


FIG. 29



NON-HARMONIC SPATIAL FREQUENCIES ALONG THE 45 DEGREE AXES, GIVING RISE TO A WEAVE-LIKE CROSS-HATCHING PATTERN IN THE SPATIAL DOMAIN

FIG. 30



NON-HARMONIC CONCENTRIC CIRCLES IN UV PLANE, WHERE PHASE HOPS QUASI-RANDOMLY ALONG EACH CIRCLE, GIVING RISE TO PSEUDO RANDOM LOOKING PATTERNS IN THE SPATIAL DOMAIN

FIG. 29A

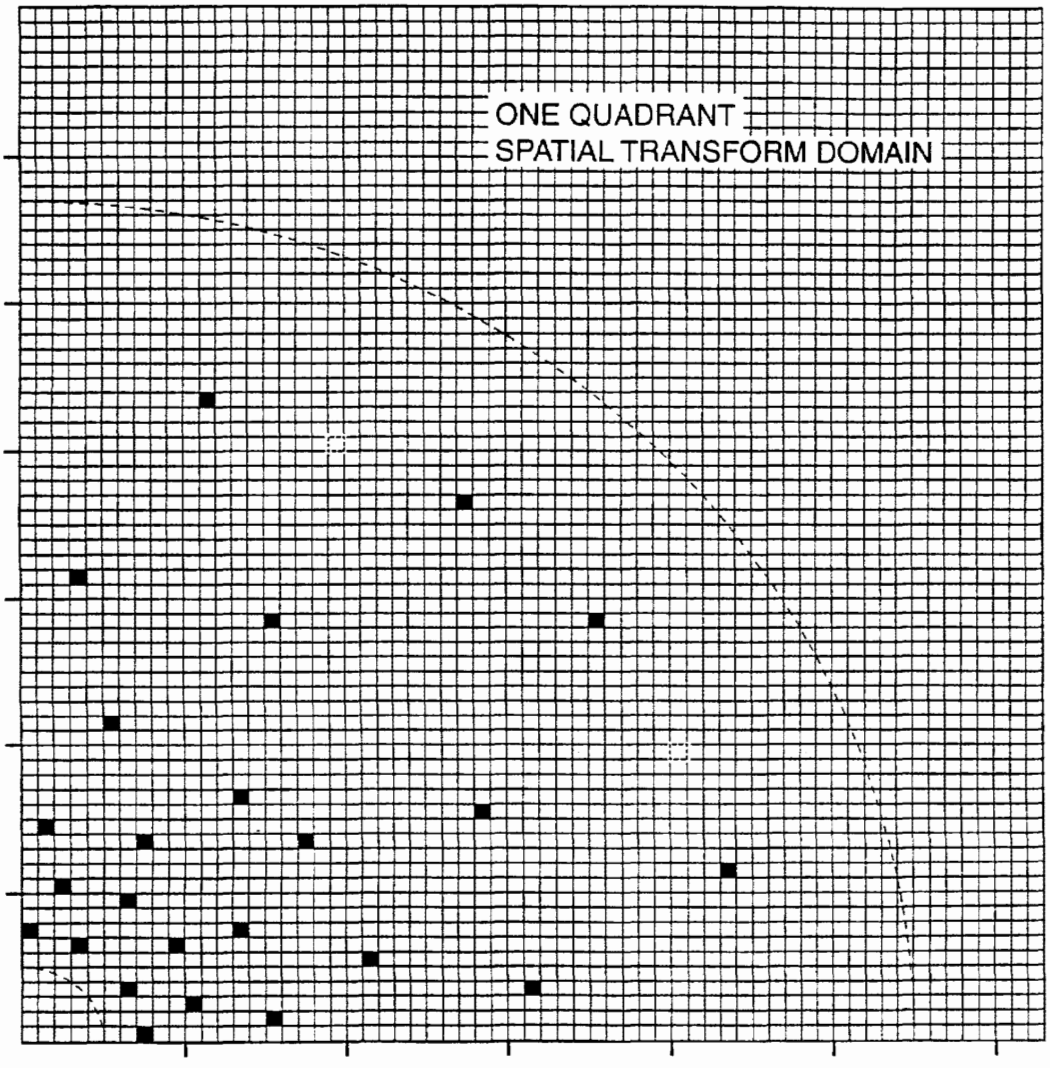
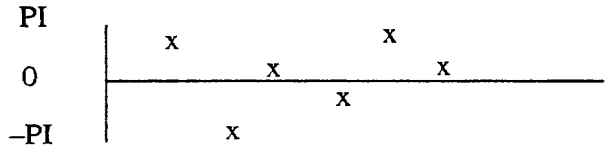
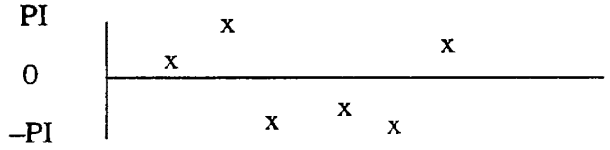


FIG. 31A



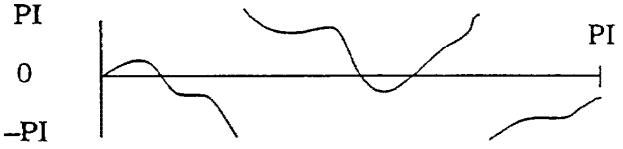
PHASE OF SPATIAL FREQUENCIES ALONG FORWARD 45 DEGREE AXES, 1008

FIG. 31B



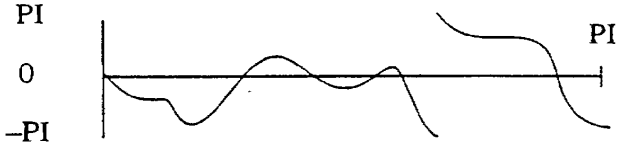
PHASE OF SPATIAL FREQUENCIES ALONG BACKWARD 45 DEGREE AXES, 1010

FIG. 32A



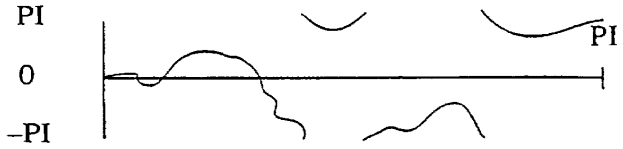
PHASE OF SPATIAL FREQUENCIES ALONG FIRST CONCENTRIC RING, 1012

FIG. 32B



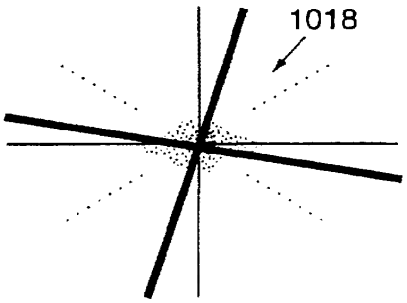
PHASE OF SPATIAL FREQUENCIES ALONG SECOND CONCENTRIC RING, 1014

FIG. 32C



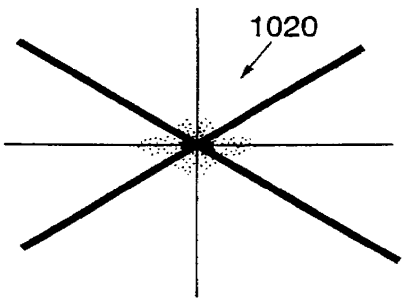
PHASE OF SPATIAL FREQUENCIES ALONG THIRD CONCENTRIC RING, 1016

FIG. 33A



ANGLE A

FIG. 33B



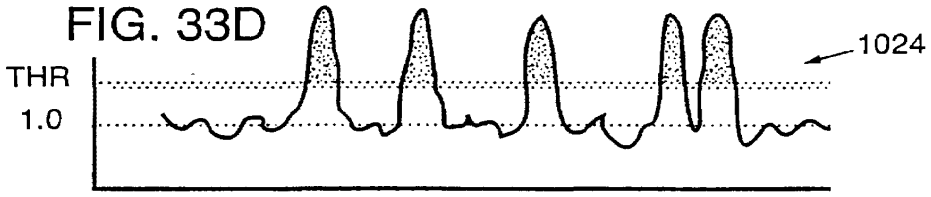
ANGLE B

FIG. 33C



POWER PROFILE ALONG ANGLE A, AS NORMALIZED BY ITS OWN MOVING AVERAGE; ONLY A MINIMAL AMOUNT EXCEEDS THRESHOLD, GIVING A SMALL INTEGRATED VALUE

FIG. 33D



POWER PROFILE ALONG ANGLE B, AS NORMALIZED BY ITS OWN MOVING AVERAGE; THIS FINDS STRONG ENERGY ABOVE THE THRESHOLD

FIG. 33E

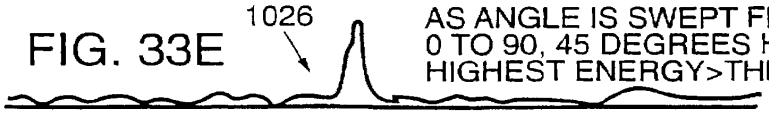


FIG. 34A

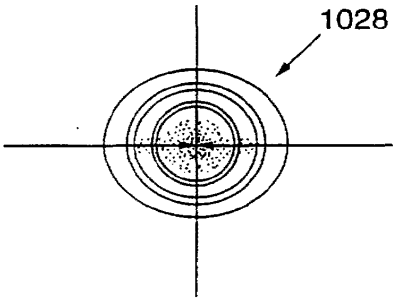


FIG. 34B

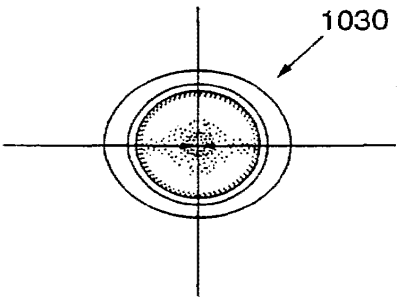


FIG. 34C

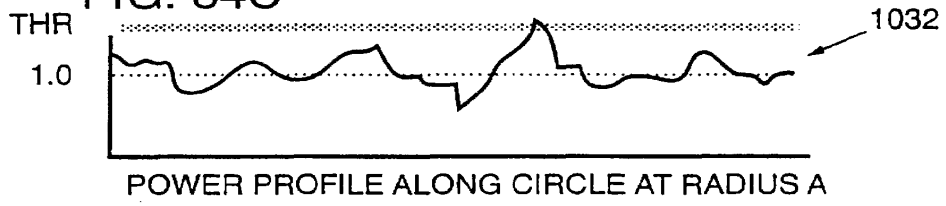


FIG. 34D

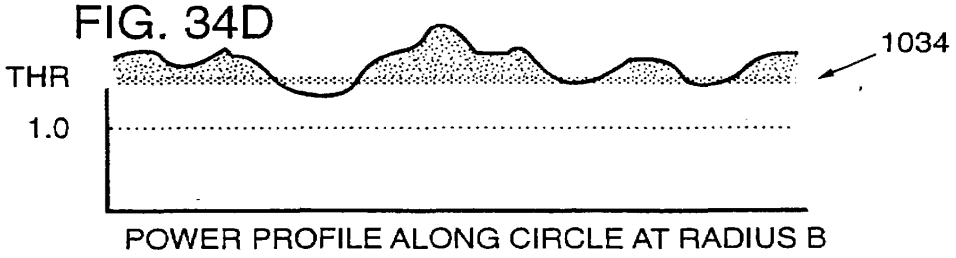


FIG. 34E

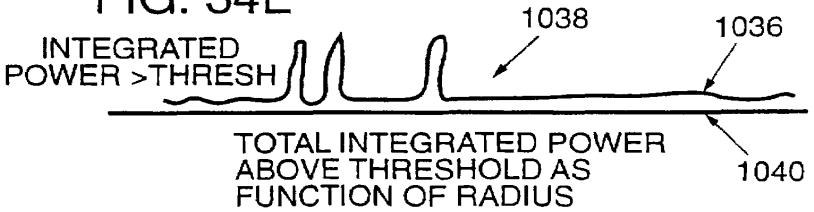
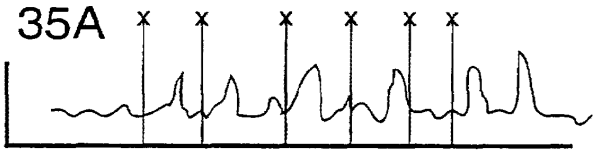
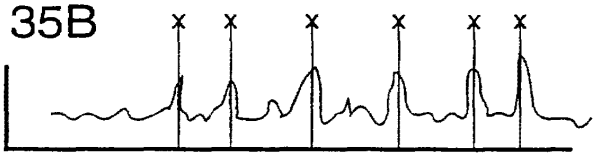


FIG. 35A



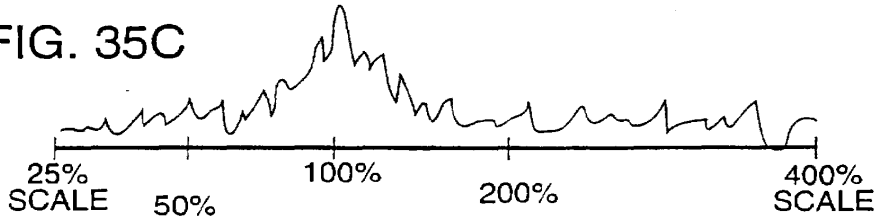
SCALE = A; ADD ALL POWER VALUES AT THE "KNOWN" FREQUENCIES", 1042

FIG. 35B



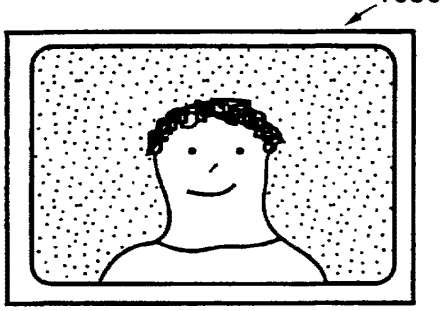
SCALE = B; ADD ALL POWER VALUES AT THE "KNOWN" FREQUENCIES", 1044

FIG. 35C



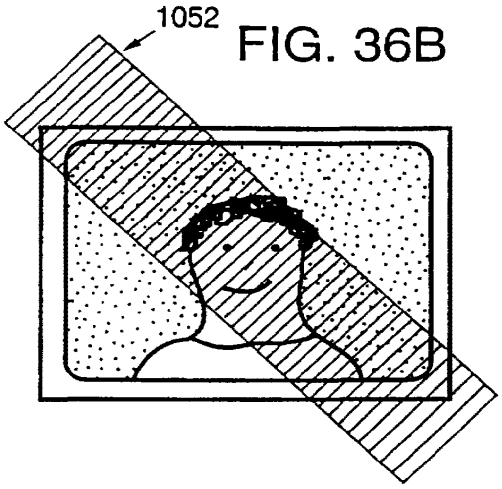
"SCALED-KERNEL" BASED MATCHED FILTER; PEAK IS WHERE THE SCALE OF THE SUBLIMINAL GRID WAS FOUND, 1046

FIG. 36A



ARBITRARY ORIGINAL IMAGE IN WHICH SUBLIMINAL GRATICULES MAY HAVE BEEN PLACED

FIG. 36B



"COLUMN SCAN" IS APPLIED ALONG A GIVEN ANGLE THROUGH THE CENTER OF THE IMAGE

FIG. 36C

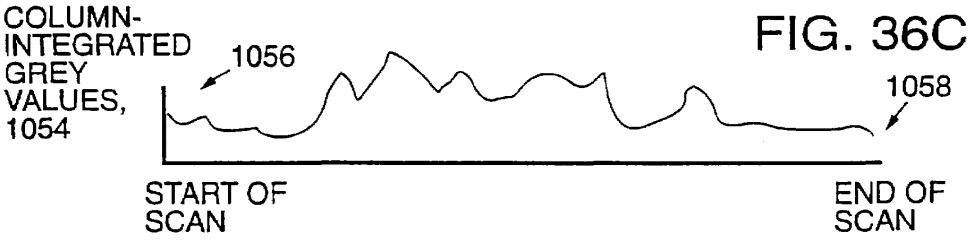
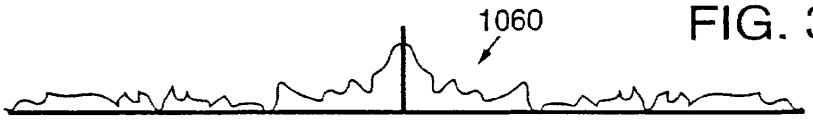


FIG. 36D



MAGNITUDE OF FOURIER TRANSFORM OF SCAN DATA

FIG. 37

PROCESS STEPS

1. SCAN IN PHOTOGRAPH
2. 2D FFT
3. GENERATE 2D POWER SPECTRUM, FILTER WITH E.G. 3X3 BLURRING KERNEL
4. STEP ANGLES FROM 0 DEGREES THROUGH 90 (1/2 DEG)
5. GENERATE NORMALIZED VECTOR, WITH POWER VALUE AS NUMERATOR, AND MOVING AVERAGED POWER VALUE AS DENOMINATOR
6. INTEGRATE VALUES AS SOME THRESHOLD, GIVING A SINGLE INTEGRATED VALUE FOR THIS ANGLE
7. END STEP ON ANGLES
8. FIND TOP ONE OR TWO OR THREE "PEAKS" FROM THE ANGLES IN LOOP 4, THEN FOR EACH PEAK...
9. STEP SCALE FROM 25% TO 400%, STEP ~1.01
10. ADD THE NORMALIZED POWER VALUES CORRESPONDING TO THE 'N' SCALED FREQUENCIES OF STANDARD
11. KEEP TRACK OF HIGHEST VALUE IN LOOP
12. END LOOP 9 AND 8, DETERMINE HIGHEST VALUE
13. ROTATION AND SCALE NOW FOUND
14. PERFORM TRADITIONAL MATCHED FILTER TO FIND EXACT SPATIAL OFFSET
15. PERFORM ANY "FINE TUNING" TO PRECISELY DETERMINE ROTATION, SCALE, OFFSET

FIG. 38

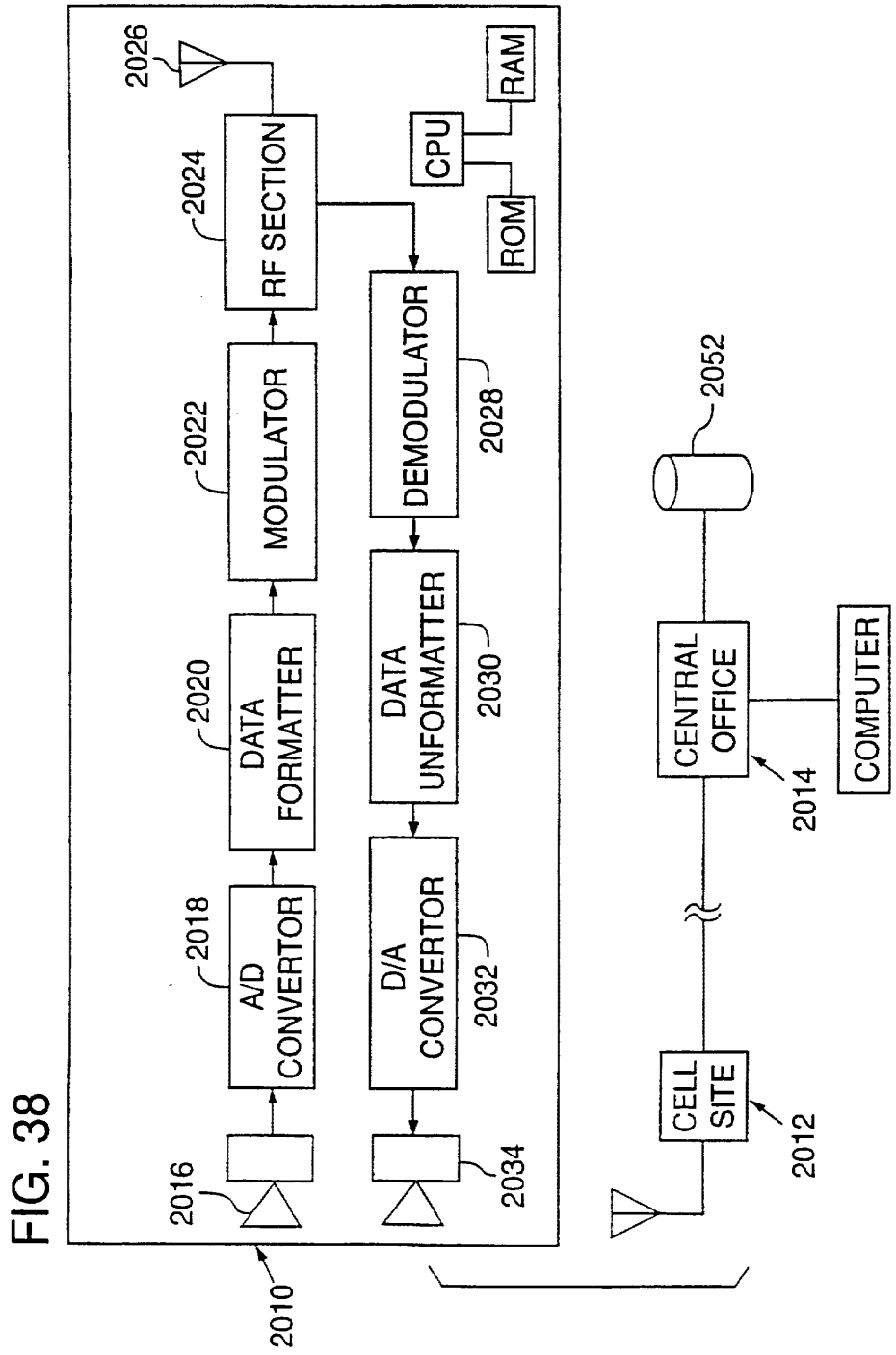
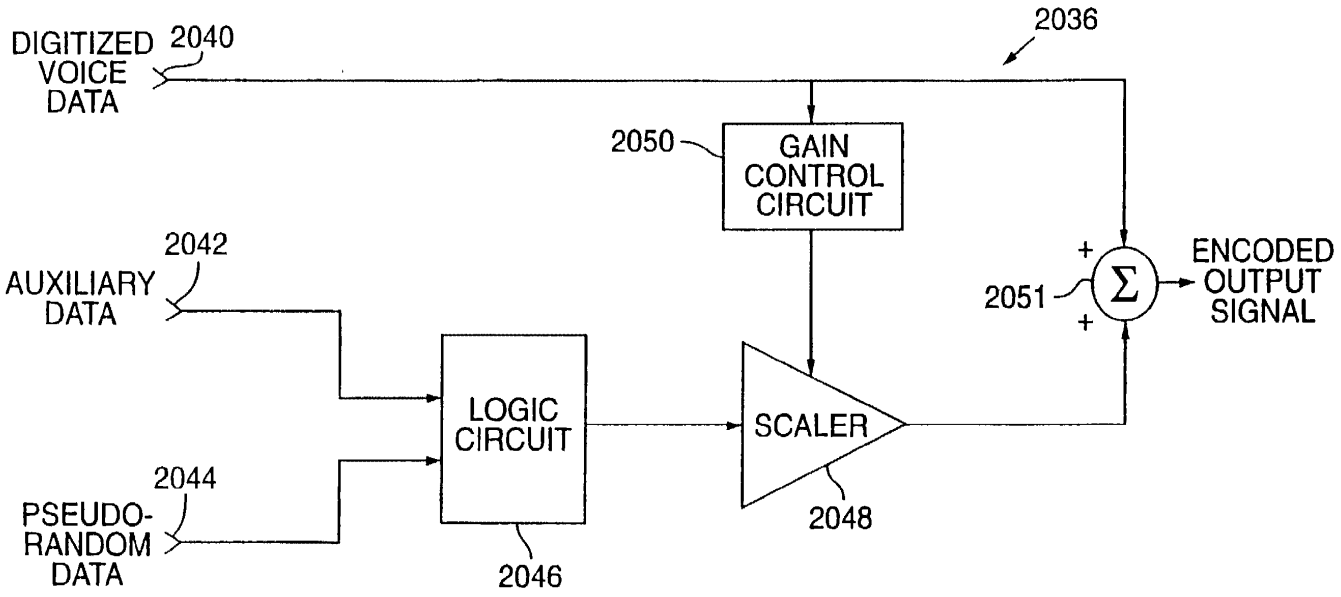

2010
2012
2014
2016
2018
2020
2022
2024
2026
2028
2030
2032
2034
2052
A/D CONVERTOR
DATA FORMATTER
MODULATOR
RF SECTION
D/A CONVERTOR
DATA UNFORMATTER
DEMODULATOR
CPU
RAM
ROM
CELL SITE
CENTRAL OFFICE
COMPUTER

FIG. 39


DIGITIZED VOICE DATA
2040
2036
2050
GAIN CONTROL CIRCUIT
2042
AUXILIARY DATA
2051
+
Σ
+
ENCODED OUTPUT SIGNAL
LOGIC CIRCUIT
SCALER
2044
PSEUDO-RANDOM DATA
2048
2046

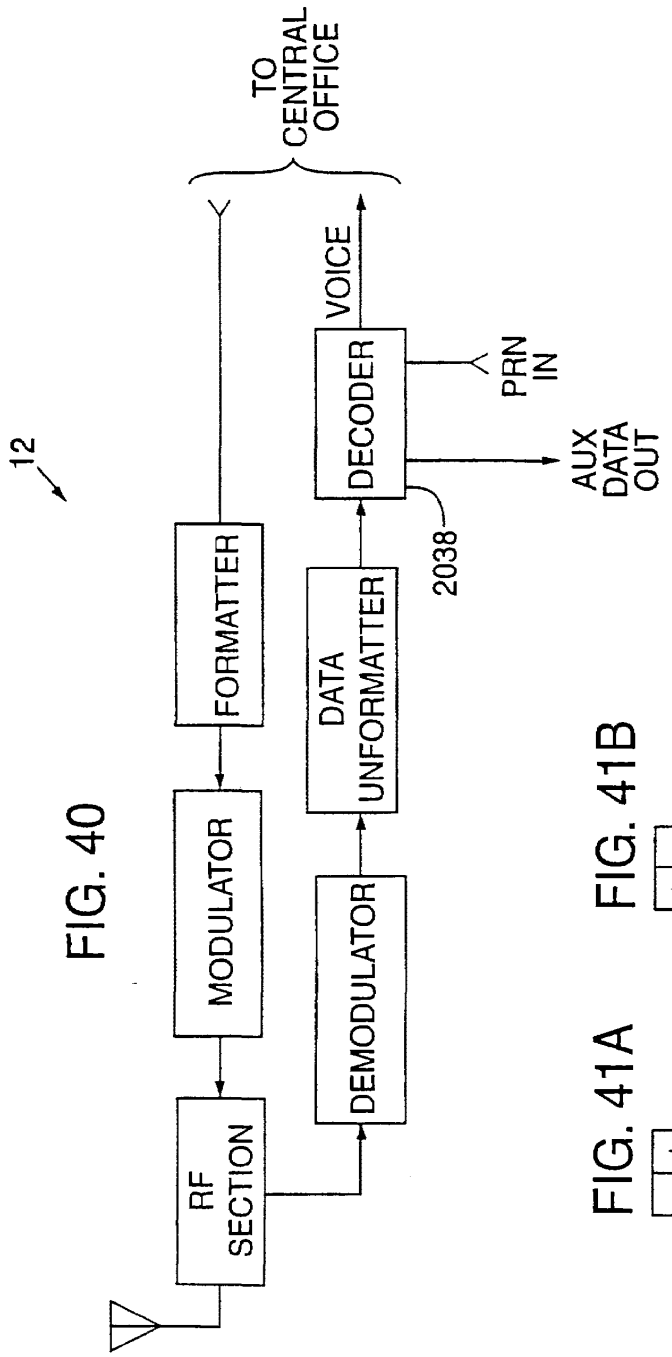


FIG. 40

FIG. 41A

+	−
−	+

FIG. 41B

−	+
+	−

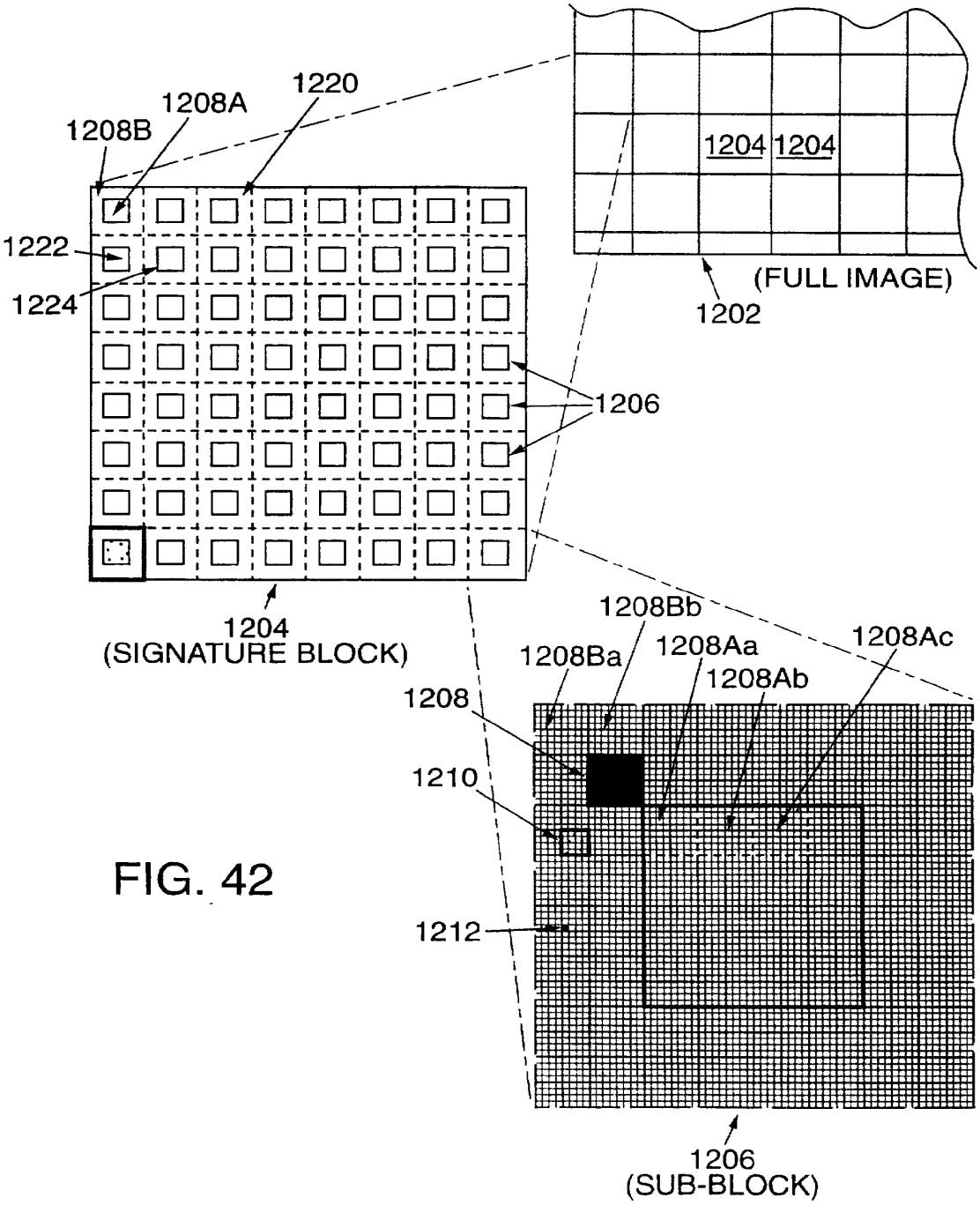

1220
1208A
1208B
1204 1204
(FULL IMAGE)
1202
1222
1224
1206
1204
(SIGNATURE BLOCK)
1208Bb
1208Ba
1208Aa
1208Ac
1208Ab
1208
1210
1212
1206
(SUB-BLOCK)

FIG. 42

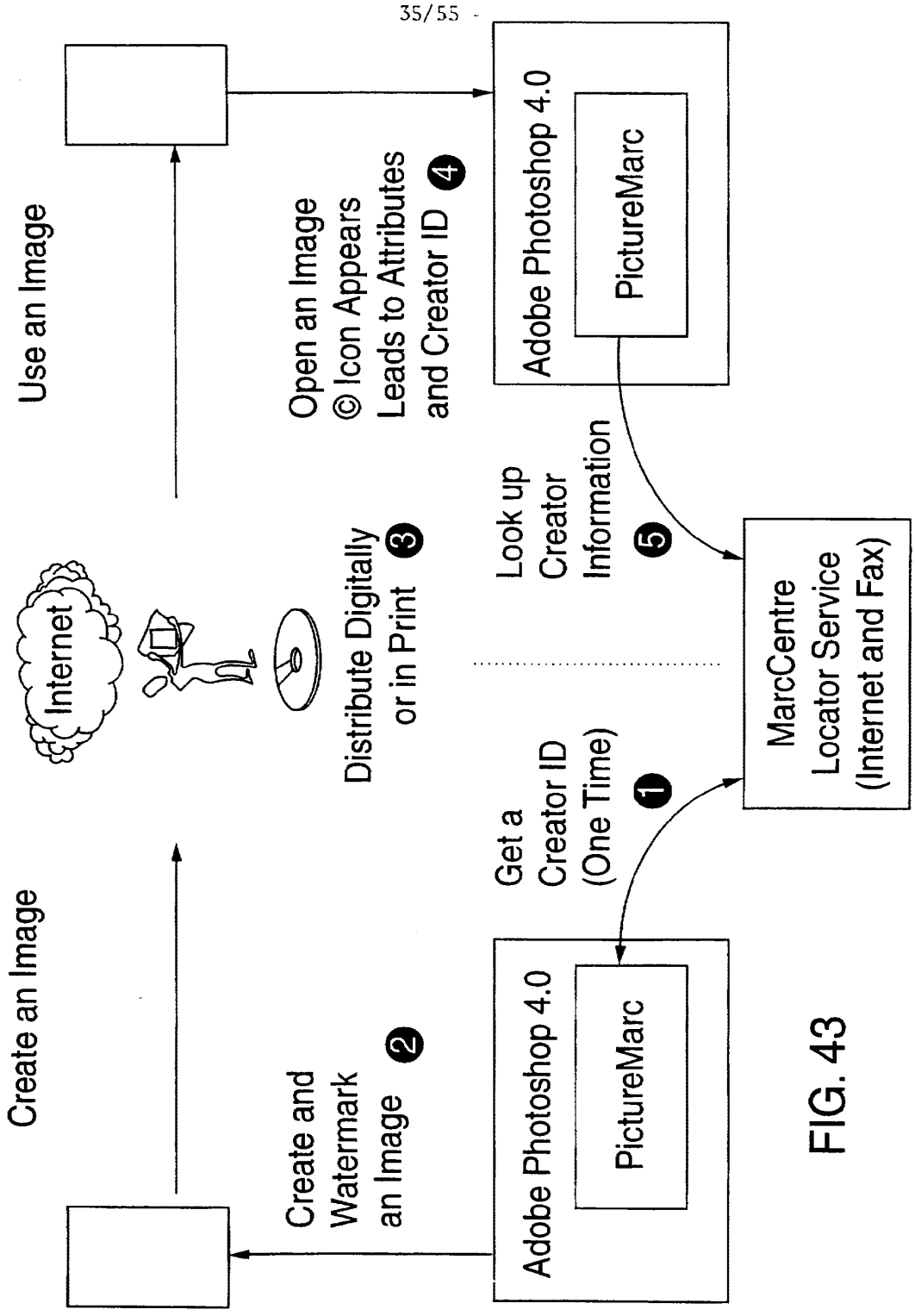

Use an Image
Open an Image
© Icon Appears
Leads to Attributes
and Creator ID 4
Adobe Photoshop 4.0
PictureMarc
Internet
Distribute Digitally
or in Print 3
Look up
Creator
Information 5
MarcCentre
Locator Service
(Internet and Fax)
Get a
Creator ID
(One Time) 1
Create an Image
Create and
Watermark
an Image 2
Adobe Photoshop 4.0
PictureMarc

FIG. 43

MarcCentre file:///C|/Data/Downloa



Welcome to MarcCentre™
Digimarc's Locator Service

MarcCentre is an on-line service which links your up-to-date contact information to every image you watermark. You provide the details you want consumers of your image to see. Each time someone reads one of your watermarked images, they can access this information about you and your work with a single mouse click.

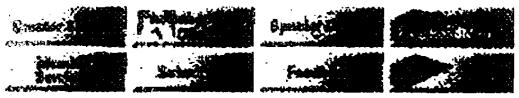
Subscribe now to give each of your images a persistent identity, communicating your copyright, and bringing consumers directly to you. Take advantage of the special offers available to you now, for a limited time.

To find out more about how MarcCentre works and how you can get started taking advantage of this locator service, browse the MarcCentre overview page.

If you are already a subscriber, you can go to Member Services to view or update your contact details.

To look up image creator/distributor information, you can search by creator ID or by other, more general, criteria.

For the technically minded, find out what's going on behind the scenes.



Digimarc Copyright Notice | About the Site

FIG. 44

Subscription : Page 1

MarcCentre™ Digimarc Corporation

Thank you for signing up with MarcCentre. There are two parts to the application form (1) your contact information and (2) Agencies or Representative contact information. Simply fill in the blanks on this page and click "Part 1 Complete" below.

Name (First/Last)*
Company
Address*
City*
State/Province*
Zip/Postal Code*
Country*
Phone Number*
Fax Number
Email Address
Your Home Page http://

You can control how much of your contact information is displayed when a potential customer finds your profile on MarcCentre. Use the selections below to tell us what you want people to see.

Phone Number	Mailing Address	Email Account
(•) Show	(•) Show	(•) Show
( ) Don't show	( ) Don't show	( ) Don't show

Yes, I want to be on your mailing lists [X]

Please select your profession (one only).

(•) Photographer
( ) Illustrator
( ) Other

In order to communicate your primary area of focus or specialty to potential customers, please select one of the options from the list below -or- type in your own in the space next to "Custom Specialty"

Photographer (none)

Illustrator (none)

Custom Specialty

FIG. 45

Subscription : Page 2 of 2

Digimarc displays an Image of the Day showcase on our web site. Would you like to participate?
◉ Yes.
○ Not at this time.
For security purposes, you will need to setup a password that allows you to edit your profile information. This password along with your Creator ID are required before you can update your contact information.

Password* Re-enter to verify*

The standard fee for the MarcCentre service is $150 (US). From now until December 31st, 1996 you can subscribe for only $79 (US), over 50% off the regular price!
For security reasons, the following credit card information is captured using secure Digimarc displays an Image of the Day showcase on our web site. Would Credit you like to participate?
◉ Visa ◉ Yes.
○ MasterCard
○ American Express

Card Number* Name Shown on Card* Expiration (MM/YY)*

When you are satisfied with your entries on this form press the "Part 1 Complete" button and continue with the second part of the subscription sign-up. If you wish to cancel, press the Back button on your browser.

Part 1 Complete

Digimarc Copyright Notice

FIG. 46

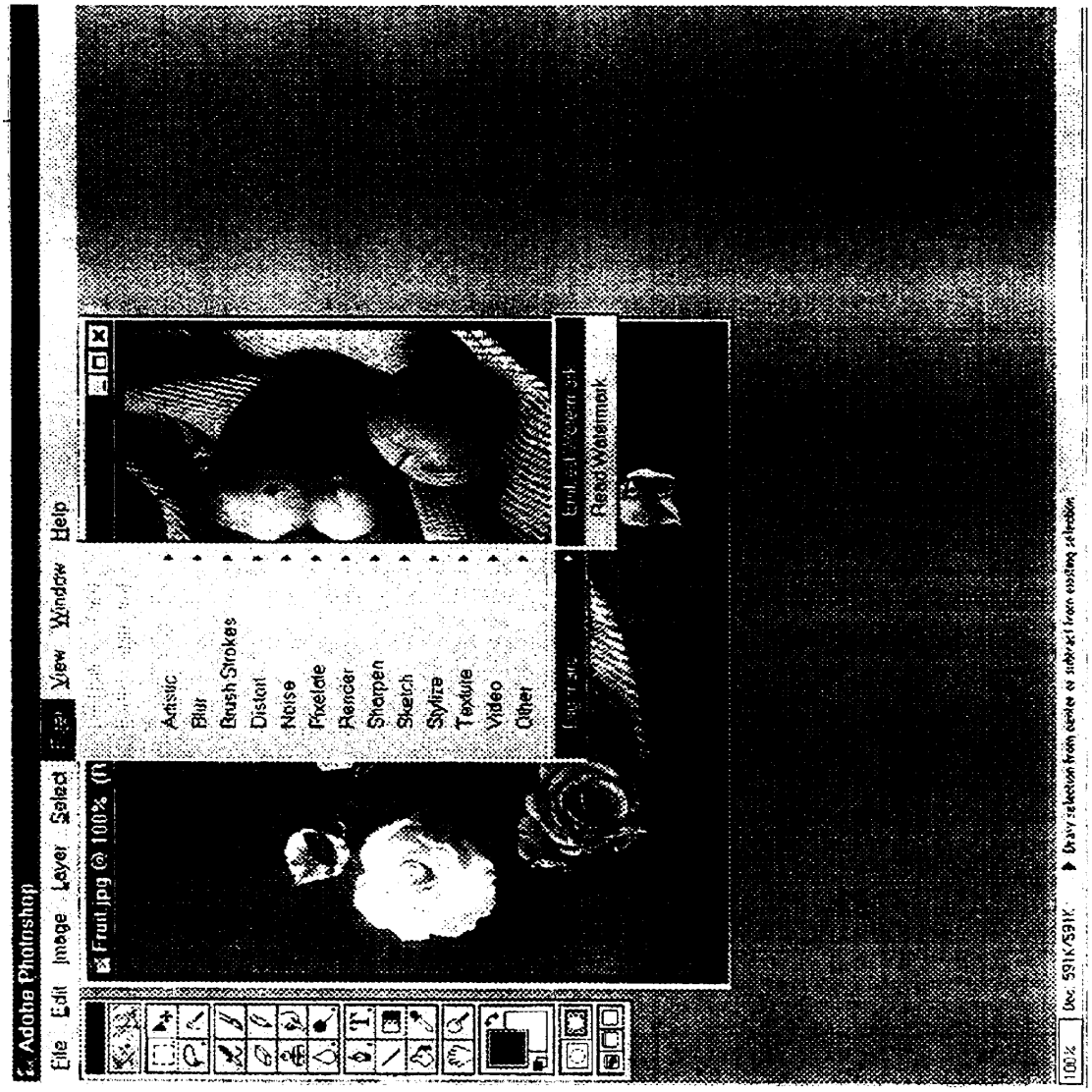

Adobe Photoshop
File Edit Image Layer Select View Window Help
Fruit.jpg @ 100% (RGB)
Artistic
Blur
Brush Strokes
Distort
Noise
Pixelate
Render
Sharpen
Sketch
Stylize
Texture
Video
Other
Read Watermark

FIG. 47


File Edit Image Layer Select Filter View Window Help
Copyright Info
Creator ID PictureMarc Demo
Personalize
Type of Use
Restricted
Royalty Free
Adult Content
Watermark Durability
less visible
more visible
less durable
more durable
OK
Cancel

FIG. 48

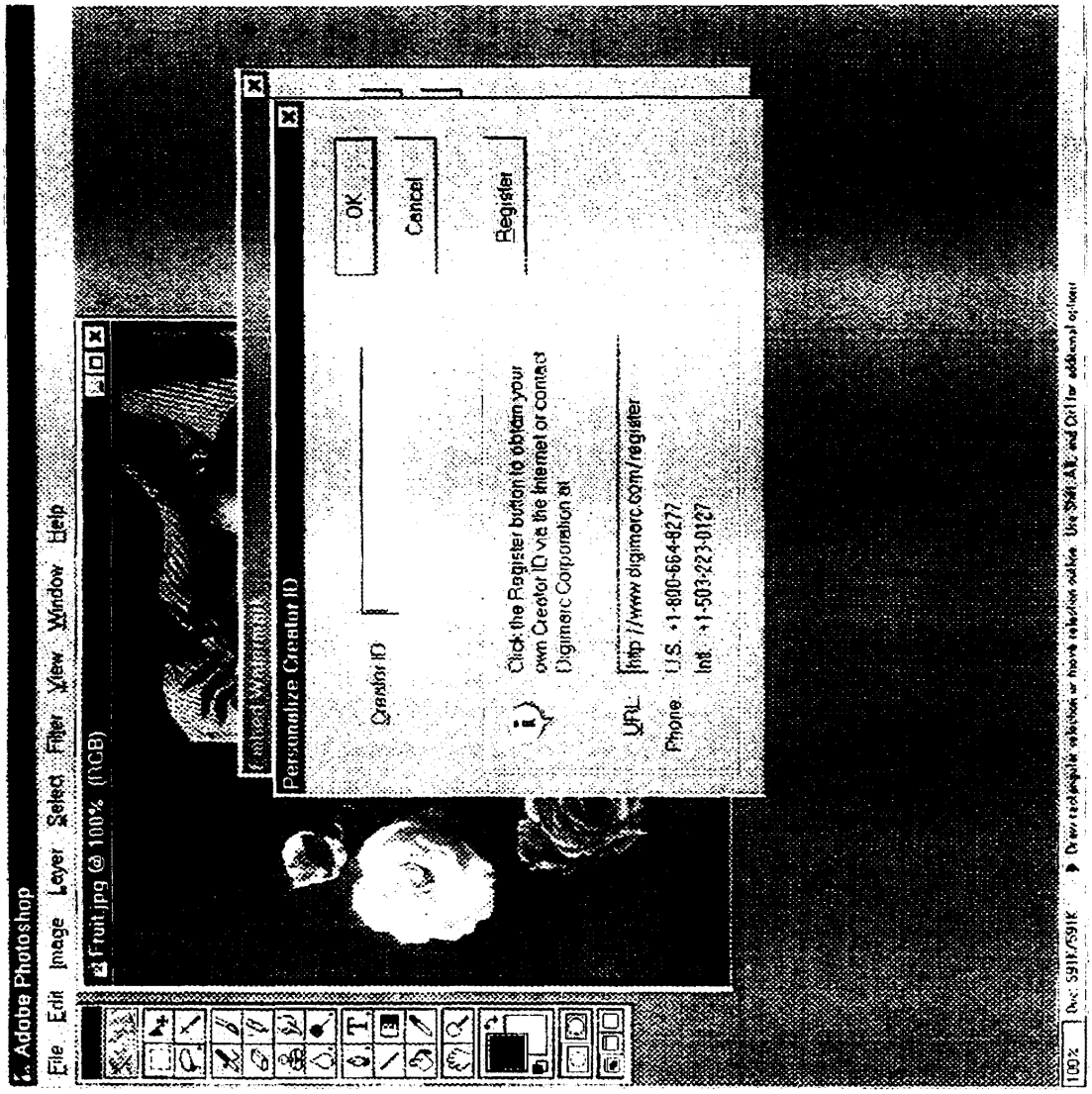

Adobe Photoshop
File Edit Image Layer Select Filter View Window Help
Fruit.jpg @ 100% (RGB)
Personalize Creator ID
Creator ID
OK
Cancel
Register
Click the Register button to obtain your own Creator ID via the Internet or contact Digimarc Corporation at
URL: http://www.digimarc.com/register
Phone: U.S. +1-800-664-8277
Int'l +1-503-223-0127

FIG. 49

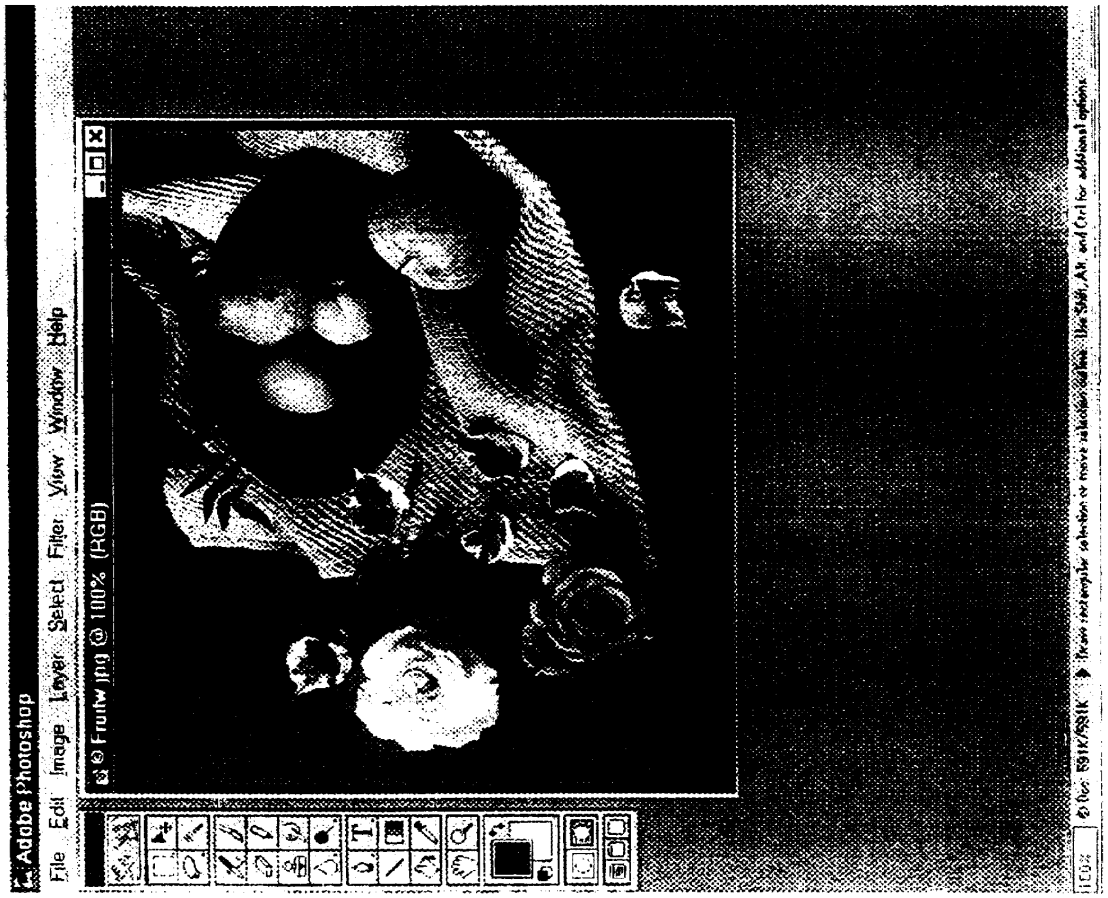

Adobe Photoshop
File Edit Image Layer Select Filter View Window Help
Fruitw.jpg @ 100% (RGB)

FIG. 50

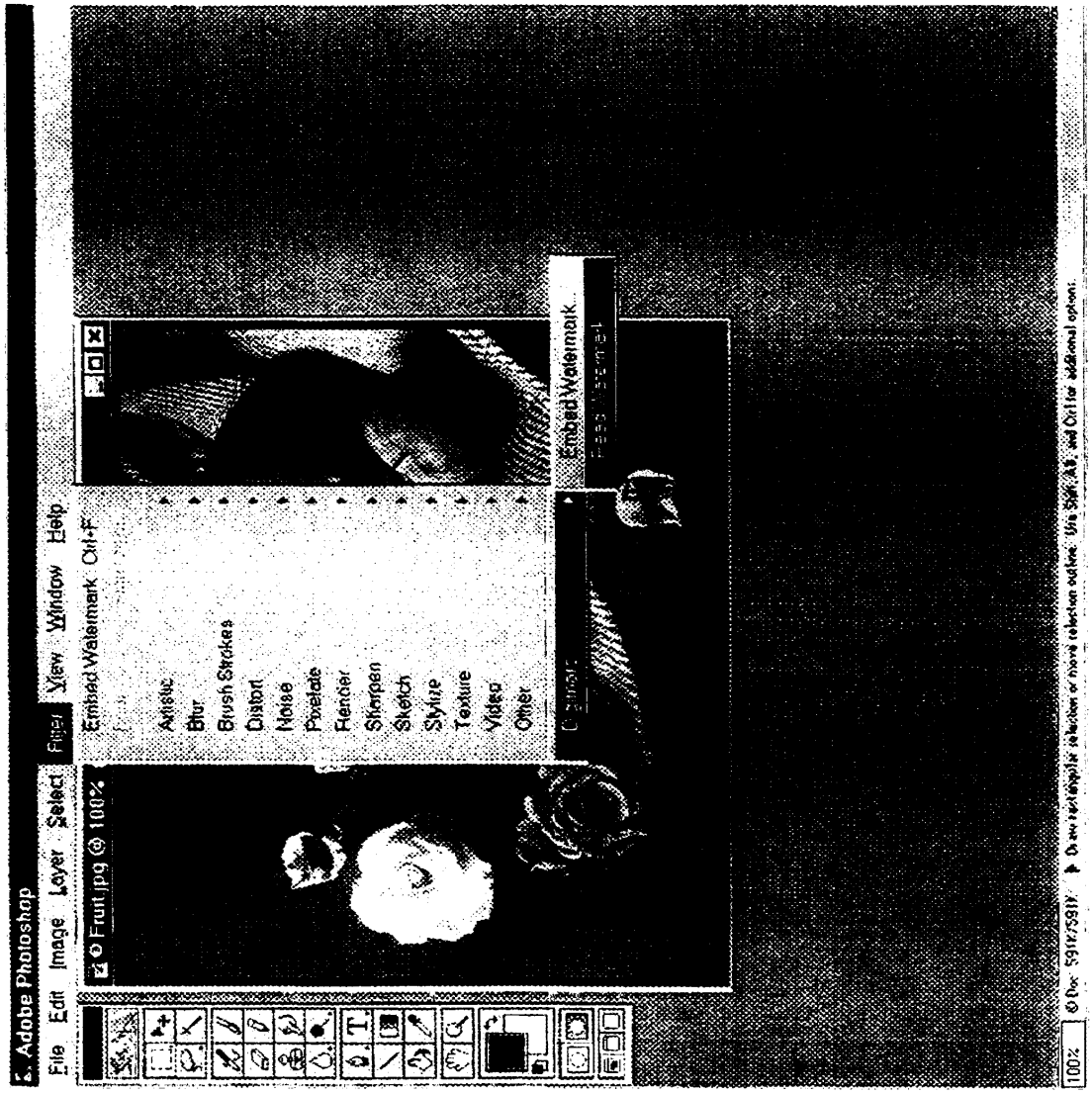

Adobe Photoshop
File Edit Image Layer Select Filter View Window Help
Fruit.jpg @ 100%
Embed Watermark Ctrl+F
Artistic
Blur
Brush Strokes
Distort
Noise
Pixelate
Render
Sharpen
Sketch
Stylize
Texture
Video
Other
Embed Watermark...
100%

FIG. 51

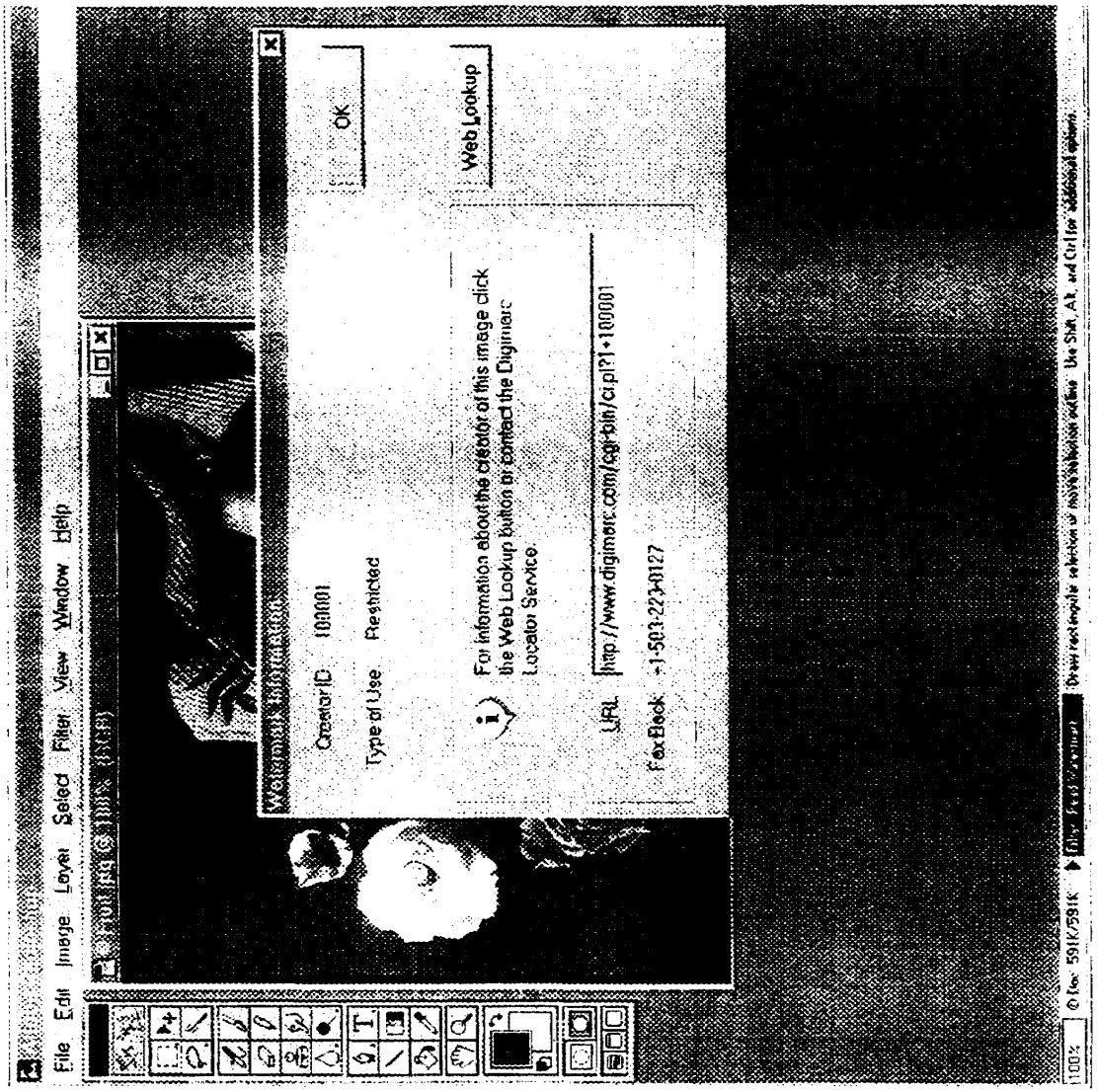

File Edit Image Layer Select Filter View Window Help
Creator ID 100001
Type of Use Restricted
For information about the creator of this image click the Web Lookup button or contact the Digimarc Locator Service.
URL http://www.digimarc.com/cgi-bin/cr.pl?1+100001
FaxBack +1-503-223-0127
OK
Web Lookup

FIG. 52

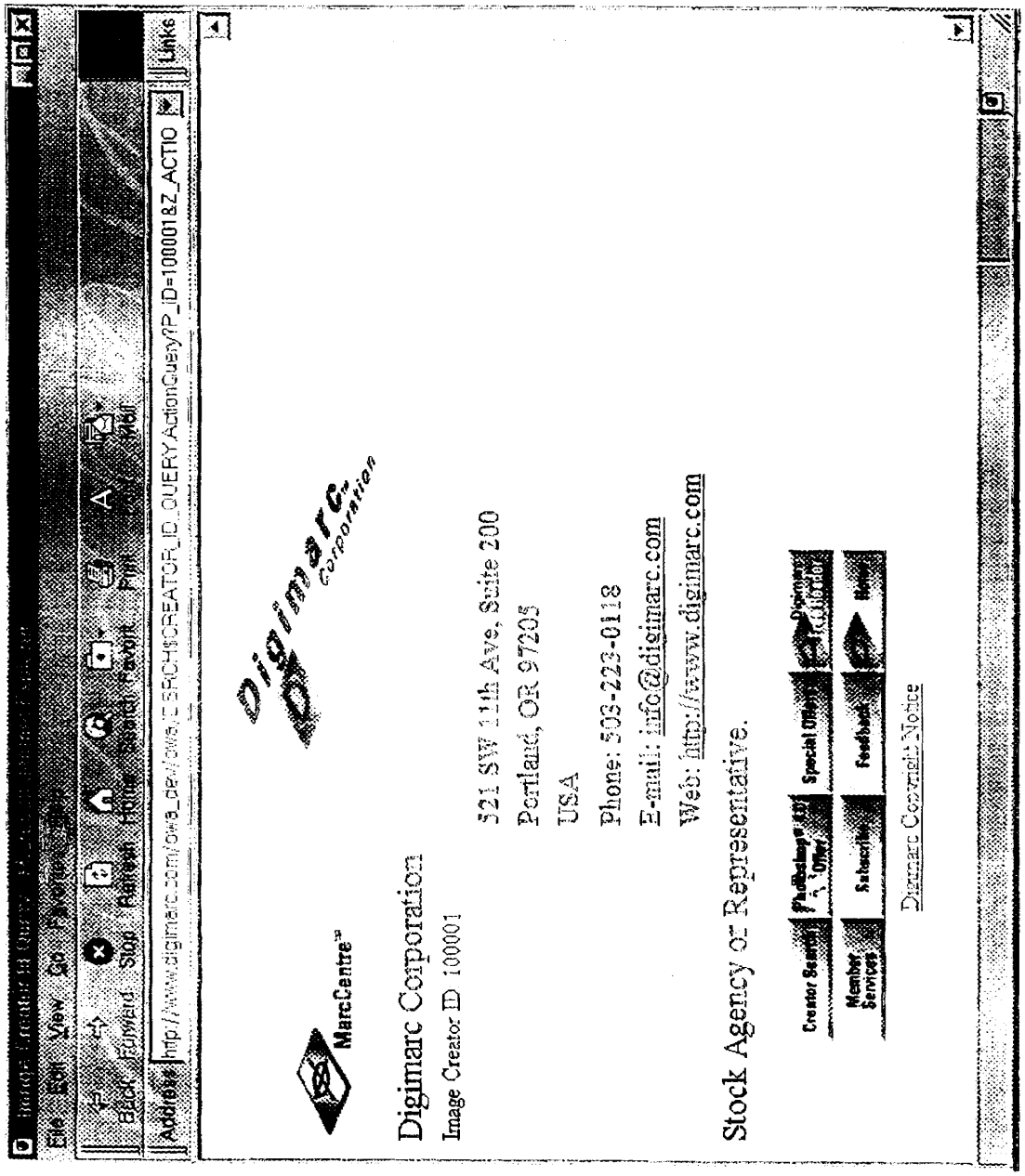
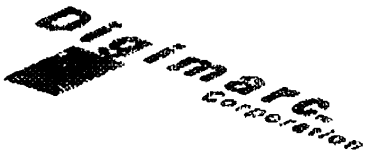

File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites Print Mail
Address http://www.digimarc.com/cgi-bin/DWB.EXE/HCREATOR_ID_QUERY.ActionQuery?P_ID=100001&Z_ACTIO
Links
MarcCentre™
Digimarc Corporation
Image Creator ID 100001
521 SW 11th Ave, Suite 200
Portland, OR 97205
USA
Phone: 503-223-0118
E-mail: info@digimarc.com
Web: http://www.digimarc.com
Stock Agency or Representative.
Creator Benefits
Special Offers
Member Services
Subscribe
Feedback
Digimarc Copyright Notice

FIG 53

Subscription : Page 1 of 1





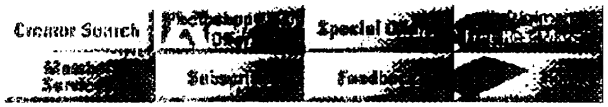
MarcCentre Image Creator Search

Enter the Image Creator ID in the box below and press "Submit Search". If it is a valid ID, the contact information details will be listed.

Image Creator ID: [ ] Submit Search

If you want to search for specific image creators by a variety of criteria such as last name, specialty, city and/or state then press the "General Search" button below.

General Search



Digimarc Copyright Notice

FIG. 54

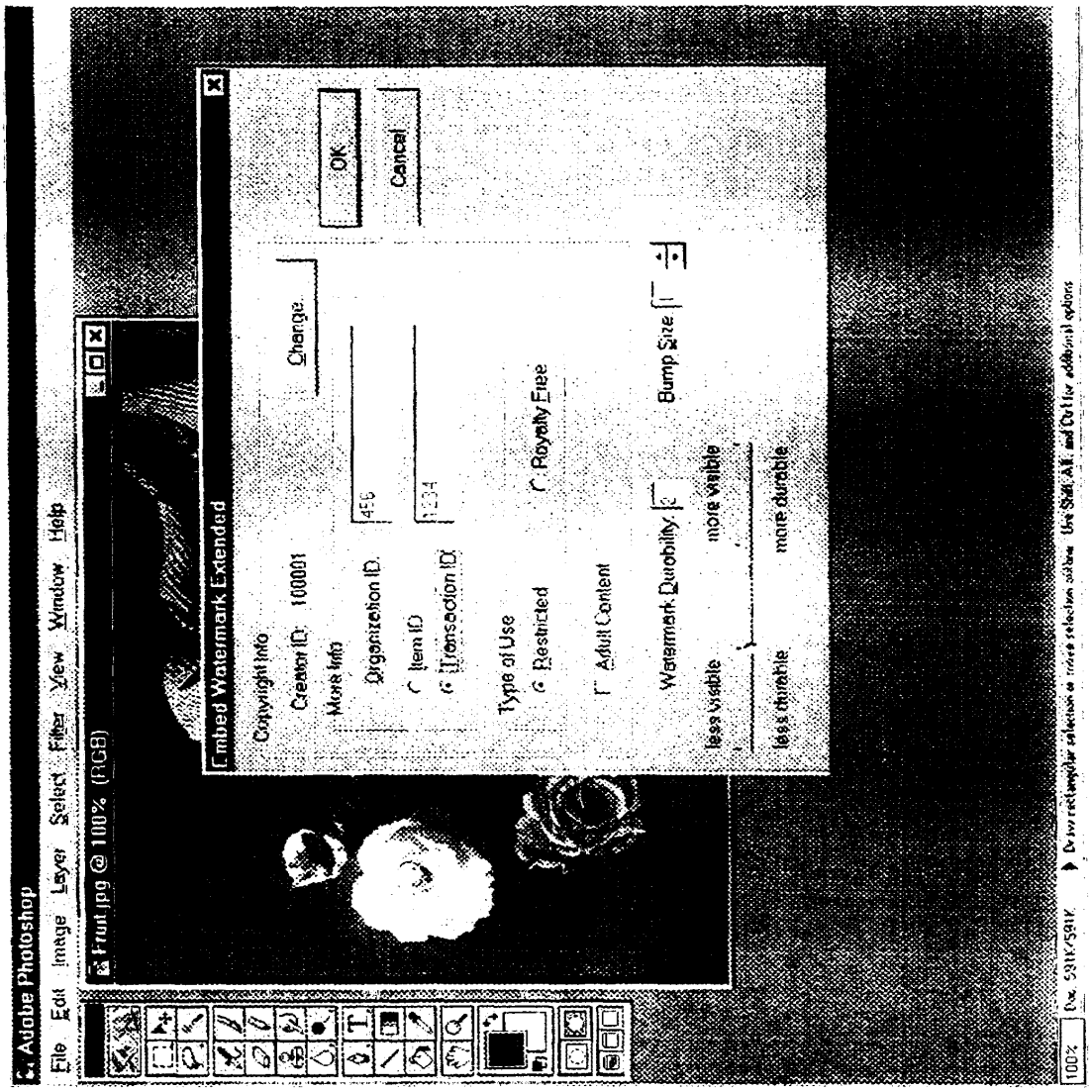

Adobe Photoshop
File Edit Image Layer Select Filter View Window Help
Fruit.jpg @ 100% (RGB)
Embed Watermark Extended
Copyright Info
Creator ID: 100001
Change...
More Info
Organization ID:
Item ID
Transaction ID:
Type of Use
Restricted
Royalty Free
Adult Content
Watermark Durability:
Bump Size
less visible
more visible
less durable
more durable
OK
Cancel

FIG. 55

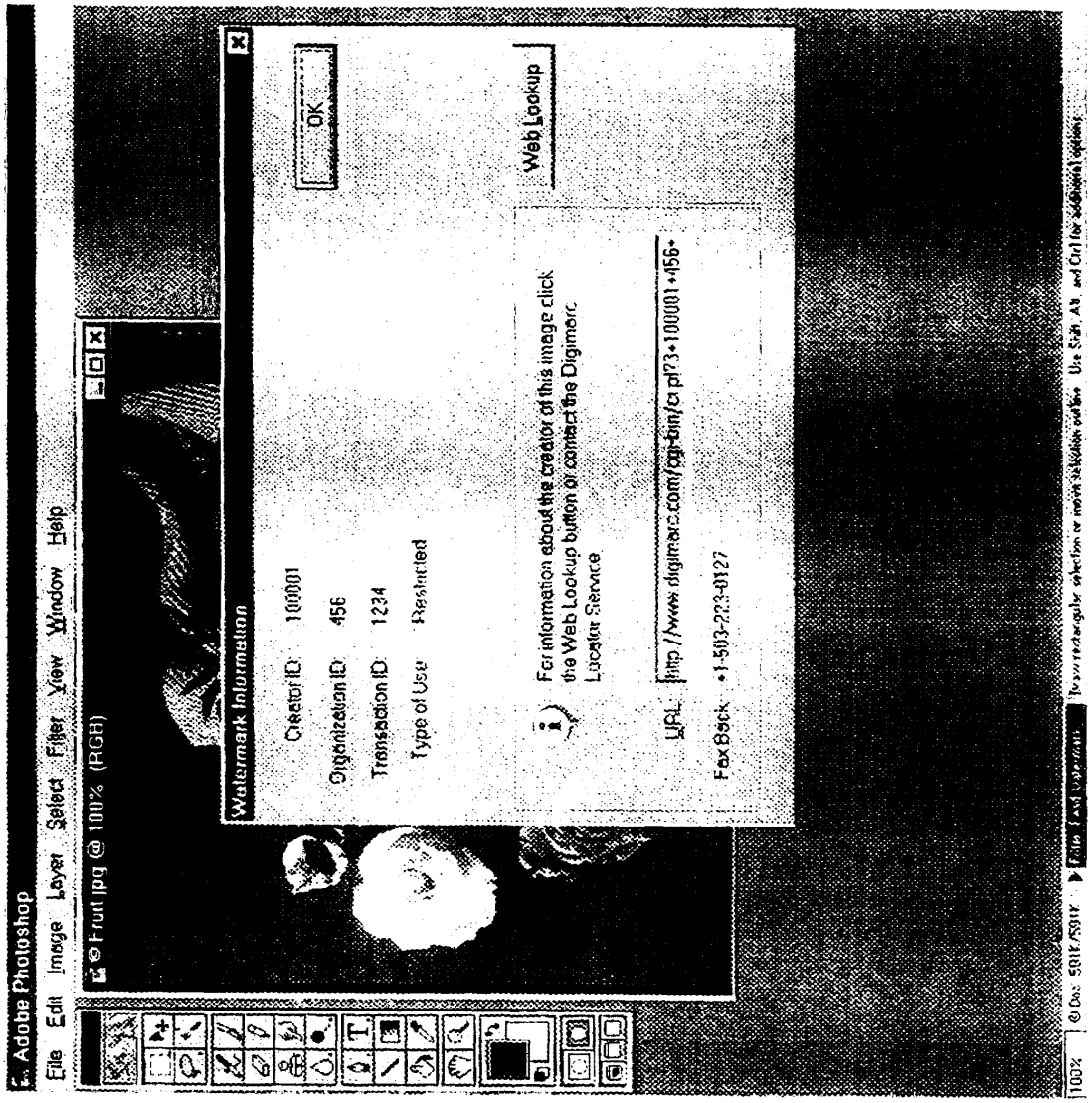

Adobe Photoshop
File Edit Image Layer Select Filter View Window Help
Fruit.jpg @ 100% (RGB)
Watermark Information
Creator ID: 100001
Organization ID: 456
Transaction ID: 1234
Type of Use: Restricted
OK
For information about the creator of this image click the Web Lookup button or contact the Digimarc Locator Service
Web Lookup
URL: http://www.digimarc.com/cgi-bin/ca.pl?3+100001+456+
Fax Back: +1-503-228-0127

FIG. 56

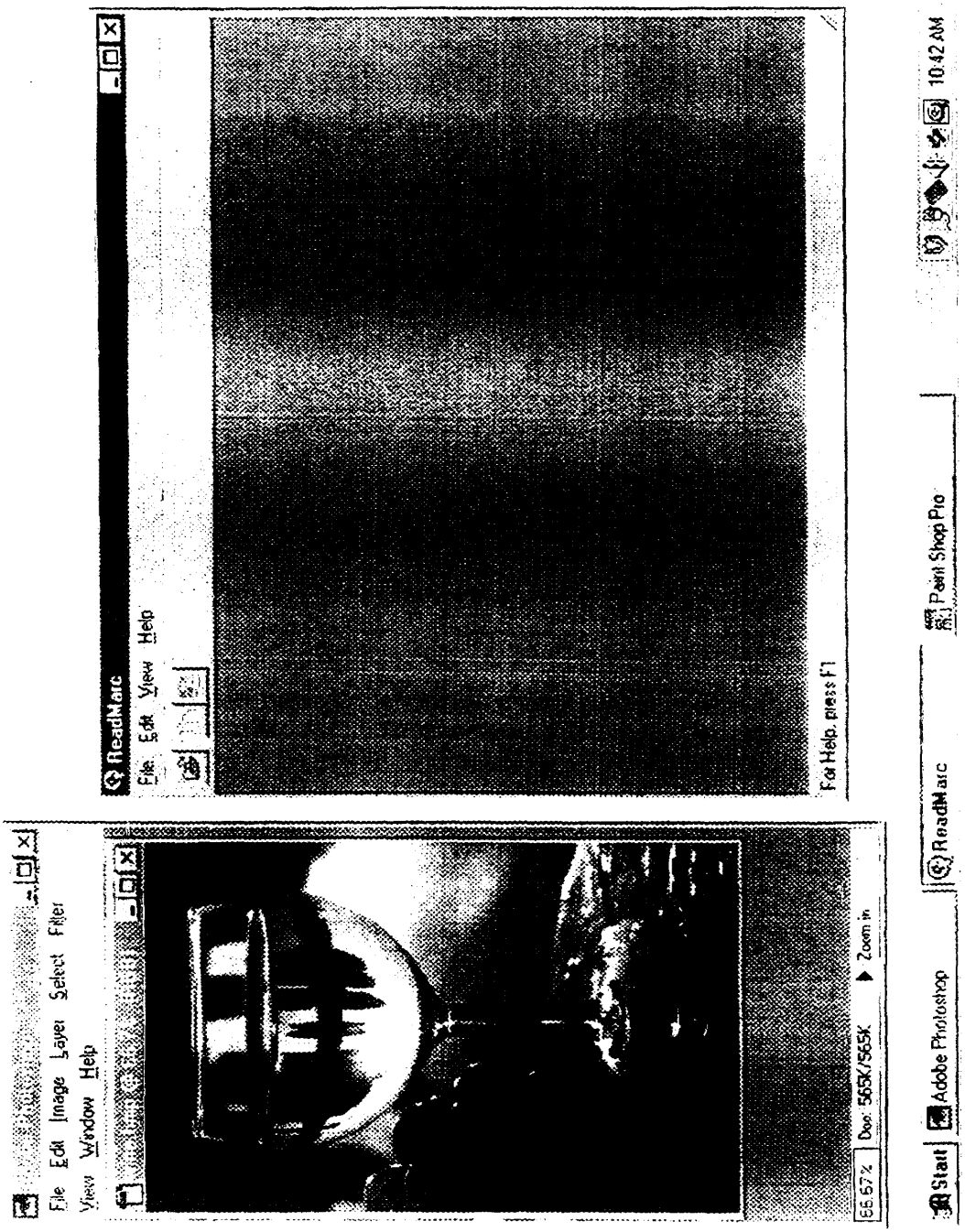

ReadMarc
File Edit View Help
For Help, press F1
File Edit Image Layer Select Filter
View Window Help
66.67%
Doc: 565K/565K
Zoom in
Start
Adobe Photoshop
ReadMarc
Paint Shop Pro
10:42 AM

FIG. 57

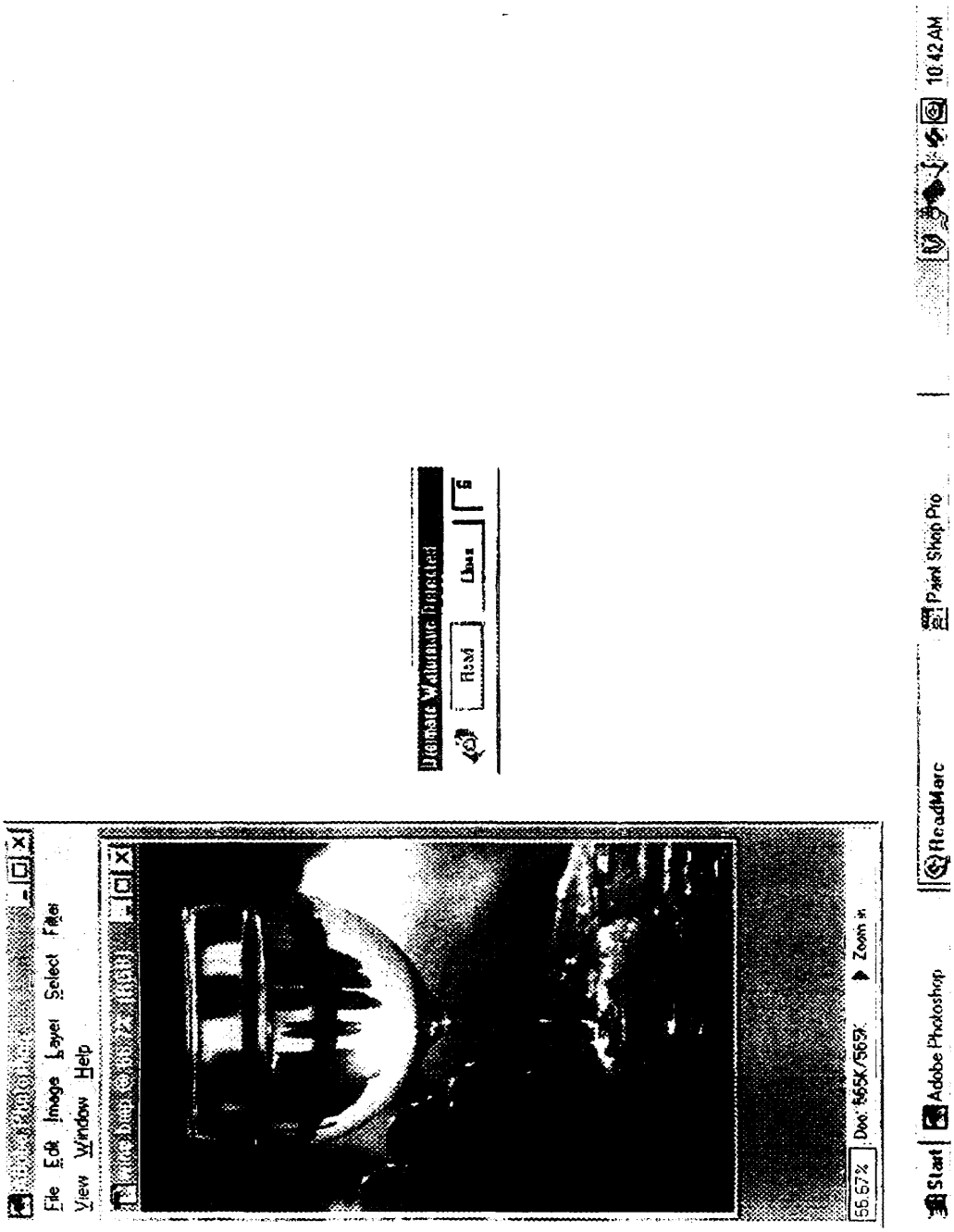

File Edit Image Layer Select Filter
View Window Help
56.67%
Doc: 565K/565K
Zoom
Dramatic Waterware Inverted
Start
Adobe Photoshop
ReadMarc
Paint Shop Pro
10:42 AM

FIG. 58

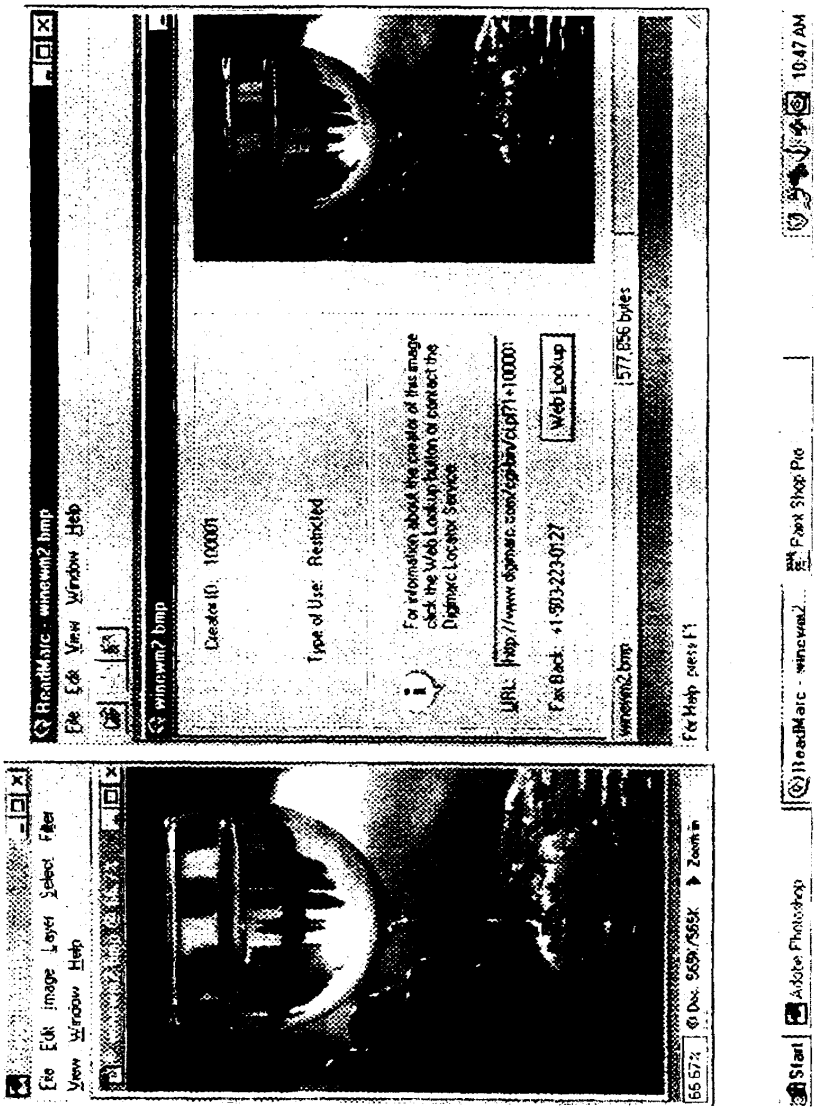

ReadMarc - winewm2.bmp
File Edit View Window Help
winewm2.bmp
Dealer ID: 10001
Type of Use: Restricted
For information about the creator of this image click the Web Lookup button or contact the Digimarc Locator Service.
URL: http://www.digimarc.com/cgi-bin/dl01?+10001
Fax Back: +1-503-234-8127
Web Lookup
577,856 bytes
winewm2.bmp
For Help, press F1
Start
Adobe Photoshop
ReadMarc - winewm2...
Paint Shop Pro
10:47 AM

FIG. 59

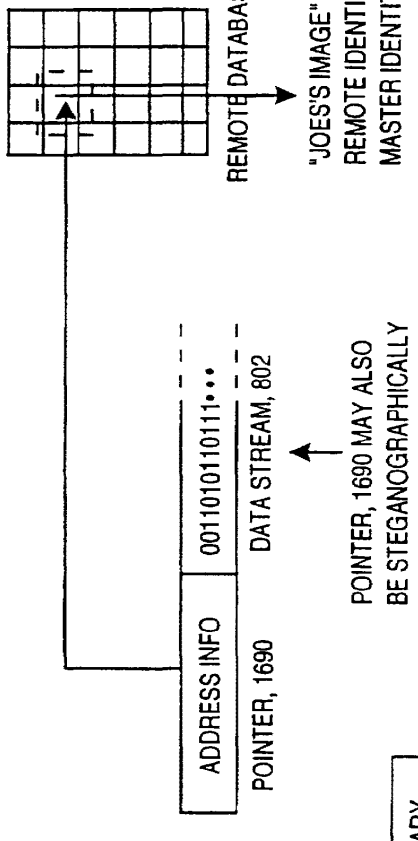

ADDRESS INFO
001101011011...
POINTER, 1690
DATA STREAM, 802
REMOTE DATABASE, 1692
"JOES'S IMAGE"
REMOTE IDENTITY,
MASTER IDENTITY, 1694
POINTER, 1690 MAY ALSO
BE STEGANOGRAPHICALLY
EMBEDDED IN DATA
STREAM IF DATA STREAM
IS SO CAPABLE
RUDIMENTARY
MASTER IDENTITY
OBJECT AND
SYSTEM

FIG. 60

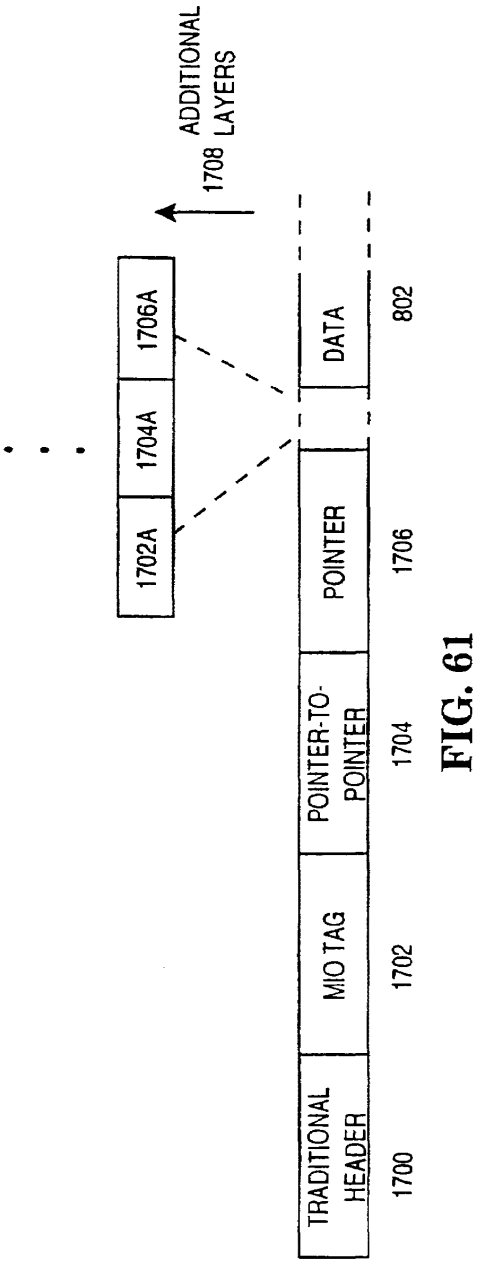

ADDITIONAL
1708 LAYERS
1702A
1704A
1706A
TRADITIONAL
HEADER
MIO TAG
POINTER-TO-
POINTER
POINTER
DATA
1700
1702
1704
1706
802

FIG. 61

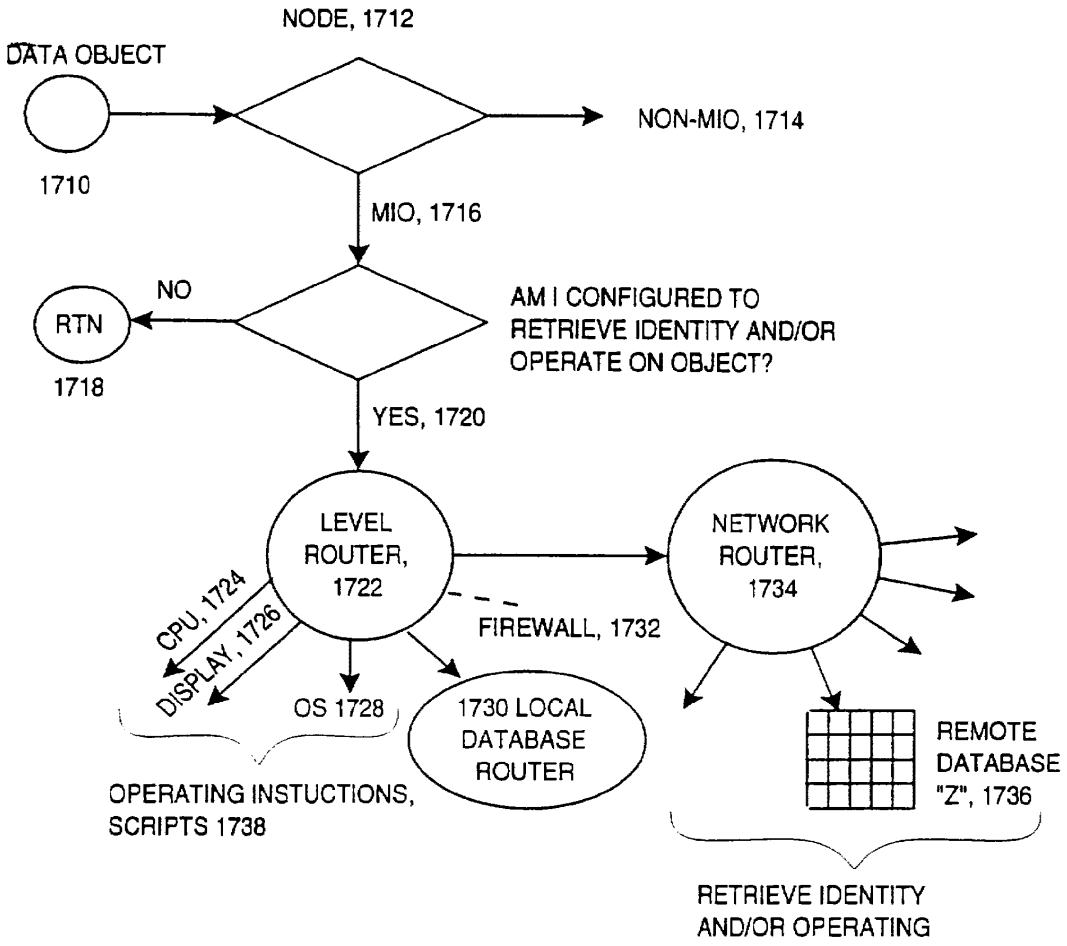

NODE, 1712
DATA OBJECT
1710
NON-MIO, 1714
MIO, 1716
NO
RTN
1718
AM I CONFIGURED TO RETRIEVE IDENTITY AND/OR OPERATE ON OBJECT?
YES, 1720
LEVEL ROUTER, 1722
NETWORK ROUTER, 1734
CPU, 1724
DISPLAY, 1726
FIREWALL, 1732
OS 1728
1730 LOCAL DATABASE ROUTER
REMOTE DATABASE "Z", 1736
OPERATING INSTUCTIONS, SCRIPTS 1738
RETRIEVE IDENTITY AND/OR OPERATING

FIG. 62

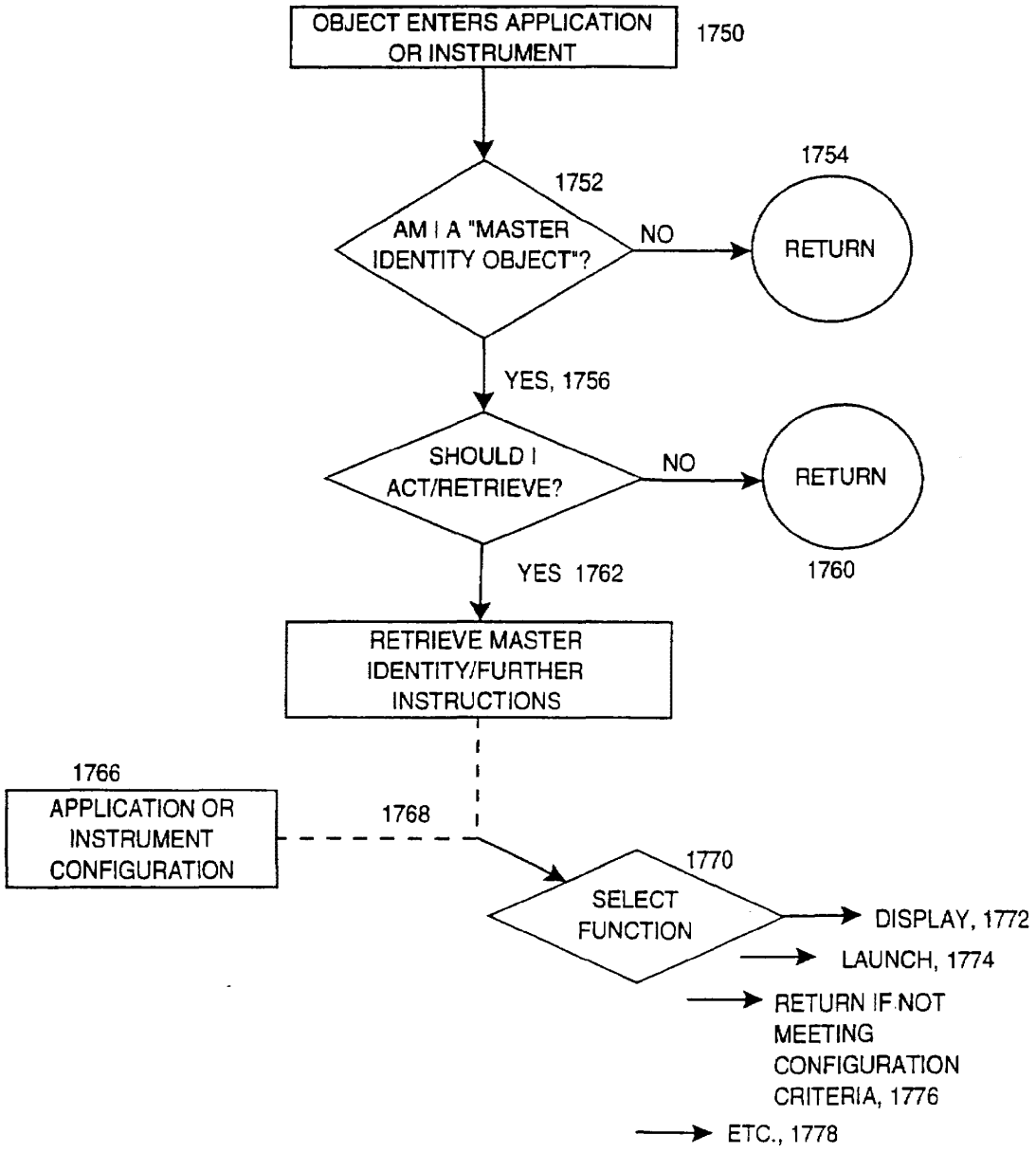

OBJECT ENTERS APPLICATION OR INSTRUMENT
1750
1752
AM I A "MASTER IDENTITY OBJECT"?
NO
1754
RETURN
YES, 1756
SHOULD I ACT/RETRIEVE?
NO
RETURN
1760
YES 1762
RETRIEVE MASTER IDENTITY/FURTHER INSTRUCTIONS
1766
APPLICATION OR INSTRUMENT CONFIGURATION
1768
1770
SELECT FUNCTION
DISPLAY, 1772
LAUNCH, 1774
RETURN IF NOT MEETING CONFIGURATION CRITERIA, 1776
ETC., 1778

FIG. 63

ATTACHED INFO
EMBEDDED SCRIPT
DATA STREAM
1800
1802
1804
ATTACHED INFO
POINTER TO SCRIPT
DATA STREAM
1800
1806
1804
REMOTE DATABASE, 1808

FIG. 64

1820
DATA OBJECT
1822
"REPORTING MODE"
MIO? 1824
YES
NO
MAY I SEND A MESSAGE TO MASTER IDENTITY? 1832
1834
YES
SEND MESSAGE TO REMOTE DATABASE, 1838
REPORT NON-HIT? 1826
NO
YES
1828
RTN
MIO SYSTEMS STATS DATABASE, 1830

FIG. 65

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US97/08351

A. CLASSIFICATION OF SUBJECT MATTER
IPC(6) :G06K 9/36
US CL :382/232
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. : 382/232, 284; 395/135; 358/454,448; 380/4,23,25,54; 283/6,93,94; 358/142,124,450; 348/475; 235/494

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

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 581 317 A2 (POWELL ET AL.) 02 FEBRUARY 1994, PAGES 1-6, FIGURES 1-5	1-5

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

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

Date of the actual completion of the international search 17 SEPTEMBER 1997	Date of mailing of the international search report 31 OCT 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 JOSE L. COUSO Toni Hill Telephone No. (703) 305-4700

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

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US97/08351

Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This international report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

Please See Extra Sheet.

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:
1-5

Remark on Protest
☐ The additional search fees were accompanied by the applicant's protest.
☐ No protest accompanied the payment of additional search fees.

Form PCT/ISA/210 (continuation of first sheet(1))(July 1992)★

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US97/08351

BOX II. OBSERVATIONS WHERE UNITY OF INVENTION WAS LACKING
This ISA found multiple inventions as follows:

This application contains the following inventions or groups of inventions which are not so linked as to form a single inventive concept under PCT Rule 13.1. In order for all inventions to be searched, the appropriate additional search fees must be paid.

Group I, claim(s) 1-5, drawn to a multi-computer system.
Group II, claim(s) 6-13, drawn to a sub-system for embedding information.
Group III, claim(s) 14, drawn to an image processing method.

The inventions listed as Groups I-III do not relate to a single inventive concept under PCT Rule 13.1 because, under PCT Rule 13.2, they lack the same or corresponding special technical features for the following reasons: The claims of these three groups are directed to different inventions which are not so linked as to form a single general inventive concept since they recite different technical features. The inventions are not linked in operation and perform completely different operations, such as Group I, a multi-computer system, Group II, a system for embedding information, and Group III, an image processor.

Since applicant has not exercised the option, the International Preliminary Authority will establish the international preliminary examination report on those parts of the international application indicated below which are claimed first, namely Claims 1-5.

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : **G06F 17/30**	**A1**	**(11) International Publication Number:** **WO 96/26494** **(43) International Publication Date:** 29 August 1996 (29.08.96)

(21) International Application Number: PCT/US96/02357

(22) International Filing Date: 22 February 1996 (22.02.96)

(30) Priority Data:
08/392,624 23 February 1995 (23.02.95) US

(71) Applicant: THE REGENTS OF THE UNIVERSITY OF CALIFORNIA [US/US]; Business and Patent Law, Mail Stop D412, Los Alamos, NM 87545 (US).

(71)(72) Applicants and Inventors: SANDFORD, Maxwell, T., II [US/US]; 160 Monte Rey Street, Los Alamos, NM 87544 (US). HANDEL, Theodore, G. [US/US]; 316 Bryce Avenue, Los Alamos, NM 87544 (US).

(74) Agents: WYRICK, Milton, D. et al.; Los Alamos National Laboratory, Mail Stop D412, Los Alamos, NM 87545 (US).

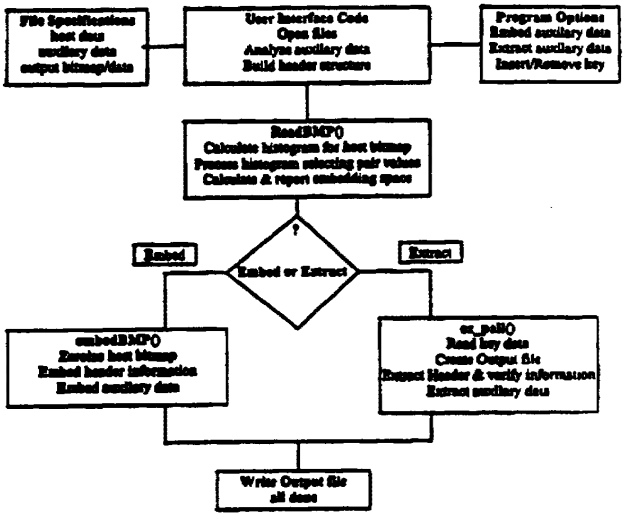
(81) Designated States: AM, AT, AU, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LT, LU, LV, MD, MG, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TT, UA, UG, UZ, VN, ARIPO patent (KE, LS, MW, SD, SZ, UG), European patent (AT, BE, CH, DE, DK, ES, 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: DATA EMBEDDING

(57) Abstract

A method of embedding auxiliary information into a set of host data, such as a photograph, television signal, facsimile transmission, or identification card. All such host data contain intrinsic noise, allowing pixels in the host data which are nearly identical and which have values differing by less than the noise value to be manipulated and replaced with auxiliary data. As the embedding method does not change the elemental values of the host data, the auxiliary data do not noticeably affect the appearance or interpretation of the host data. By a substantially reverse process, the embedded auxiliary data can be retrieved easily by an authorized user.



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

DATA EMBEDDING

FIELD OF THE INVENTION

The present invention generally relates to digital manipulation of numerical data and, more specifically, to the embedding of external data into existing data fields. This invention was made with Government support under Contract No. W-7405-ENG-36 awarded by the U.S. Department of Energy. The Government has certain rights in the invention.

The use of data in digital form is revolutionizing communication throughout the world. Much of this digital communication is over wire, microwaves, and fiber optic media. Currently, data can be transmitted flawlessly over land, water, and between satellites. Satellites in orbit allow communication virtually between any two points on earth, or in space.

In many situations, it may be of benefit to send particular secondary data along with the primary data. Secondary data could involve the closed captioning of television programs, identification information associated with photographs, or the sending of covert information with facsimile transmissions. Such a technique is suited also for use as a digital signature verifying the origin and authenticity of the primary data.

Data in digital form are transmitted routinely using wide band communications channels. Communicating in digital fashion is facilitated greatly by error-correcting software and hardware protocols that provide absolute data fidelity. These communication systems ensure that the digital bit stream transmitted by one station is received by the other station unchanged.

However, most digital data sources contain redundant information and intrinsic noise. An example is a digital image generated by scanning a photograph, an original work of electronic art, or a digitized video signal. In the scanning or digital production process of such images, noise is introduced in the digital rendition. Additionally, image sources, such as photographic images and identification cards, contain noise resulting from the grain structure of the film, optical aberrations, and subject motion. Works of art contain noise which is introduced by brush strokes, paint texture, and artistic license.

Redundancy is intrinsic to digital image data, because any particular numerical value of the digital intensity exists in many different parts of the image. For example, a given grey-level may exist in the image of trees, sky, people or other objects. In any digital image, the same or similar numerical picture element, or pixel value, may represent a variety of image content. This means that pixels having similar numerical values and frequency of occurrence in different parts of an image can be interchanged freely, without noticeably altering the appearance of the image or the statistical frequency of occurrence of the pixel values.

Redundancy also occurs in most types of digital information, whenever the same values are present more than once in the stream of numerical values representing the information. For a two-color, black and white FAX image, noise consists of the presence or absence of a black or white pixel value. Documents scanned into black and white

BITMAP® format contain runs of successive black (1) and white (0) values. Noise in these images introduces a variation in the length of a pixel run. Runs of the same value are present in many parts of the black and white image, in different rows. This allows the present invention also to be applied to facsimile transmissions.

The existence of noise and redundant pixel information in digital data permits a process for implanting additional information in the noise component of digital data. Because of the fidelity of current digital communication systems, the implanted information is preserved in transmission to the receiver, where it can be extracted. The embedding of information in this manner does not increase the bandwidth required for the transmission because the data implanted reside in the noise component of the host data. One may convey thereby meaningful, new information in the redundant noise component of the original data without it ever being detected by unauthorized persons.

It is therefore an object of the present invention to provide apparatus and method for embedding data into a digital information stream so that the digital information is not changed significantly.

It is another object of the present invention to provide apparatus and method for thwarting unauthorized access to information embedded in normal digital data.

Additional objects, advantages and novel features of the invention will be set forth in part in the description which follows, and in part will become apparent to those skilled in the art upon examination of the following or may

be learned by practice of the invention. The objects and advantages of the invention may be realized and attained by means of the instrumentalities and combinations particularly pointed out in the appended claims.

BACKGROUND OF THE INVENTION

Additional objects, advantages and novel features of the invention will be set forth in part in the description which follows, and in part will become apparent to those skilled in the art upon examination of the following or may be learned by practice of the invention. The objects and advantages of the invention may be realized and attained by means of the instrumentalities and combinations particularly pointed out in the appended claims.

SUMMARY OF THE INVENTION

In accordance with the purposes of the present invention there is provided a method of embedding auxiliary data into host data comprising the steps of creating a digital representation of the host data consisting of elements having numerical values and containing a noise component; creating a digital representation of the auxiliary data in the form of a sequence of bits; evaluating the noise component of the digital representation of the host data; comparing elements of the host data with the noise component to determine pairs of the host elements having numerical values which differ by less than said value of said noise component; and replacing individual values of the elements with substantially equivalent values from said pairs of elements in order to embed individual bit values of the auxiliary data corresponding to the sequence of bits of

the auxiliary data; and outputting the host data with the auxiliary data embedded into the host data as a file.

In accordance with the purposes of the present invention there is further provided a method of extracting embedded auxiliary data from host data containing a noise component comprising the steps of extracting from the host data a bit sequence indicative of the embedded auxiliary data, and which allows for verification of the host data; interpreting the host data-element pairs which differ by less than the value of the noise component and which correspond to bit values of the auxiliary data;identifying the auxiliary data bit sequence corresponding to the pair values; and extracting the auxiliary data as a file.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and form a part of the specification, illustrate the embodiments of the present invention and, together with the description, serve to explain the principles of the invention. In the drawings:

FIGURE 1 is a block diagram illustrating the processes used in the embedding and extraction of data from a host.

FIGURE 2 is a partial listing of computer code used for determining host data pairs having similar values and for converting RGB components to HSI components.

FIGURE 3 is a partial listing of computer code used for eliminating duplicate host data pairs.

FIGURE 4 is a partial listing of computer code which, for Truecolor images, introduces a constraint on the frequency of occurrence of host data pairs that minimizes the effect of embedding on the host data histogram.

FIGURE 5 is a partial listing of computer code that performs the actual embedding of auxiliary data into the host data, including the considerable information which is necessary to manipulate the data in the header information, auxiliary bit-stream, and the host data files.

FIGURE 6 is a partial listing of computer code that analyzes the lengths of runs in a row of pixels in two-color facsimile host data.

FIGURE 7 is a partial listing of computer code whose purpose is to ensure that the first pixel in a PACKET-START data row starts in an even column number. The location of the first pixel in the row flags the start of the data packets.

FIGURE 8 is a partial listing of computer code for embedding data into two-color host images, such as facsimile transmissions.

DETAILED DESCRIPTION

The present invention allows data to be embedded into a digital transmission or image without naturally discernible alteration of the content and meaning of the transmission or image. This is made possible because of the technique of the present invention, in which similar pixel values in a set of digital host data are re-ordered according to the desired embedded or implanted information. The host data image examples are represented in the MICROSOFT® BITMAP® (.BMP) format, so that the resulting image contains the embedded auxiliary information without that information being readily discernible.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Ex.1002 / Page 417 of 1311
APPLE INC.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

WHAT IS CLAIMED IS:

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

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

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

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

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

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

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

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

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

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

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

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

identifying said auxiliary data using said bit sequence; and

extracting said auxiliary data as a file.

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

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

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

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

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

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

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

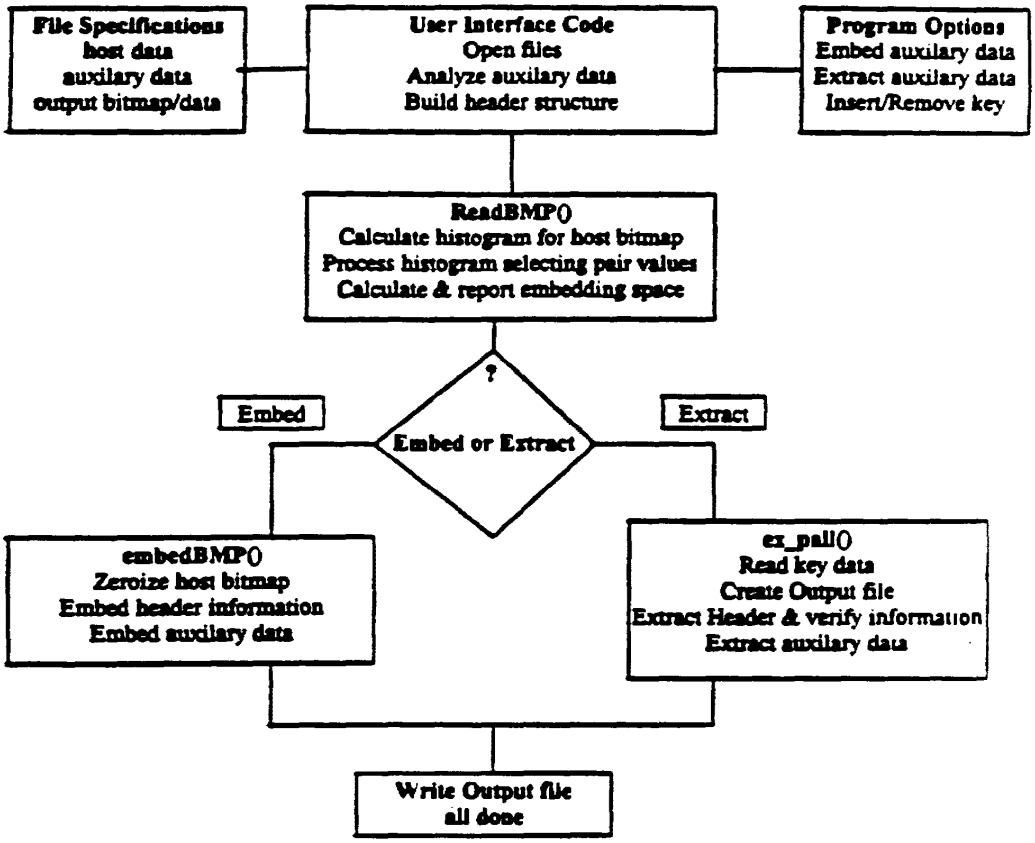

File Specifications
host data
auxilary data
output bitmap/data
User Interface Code
Open files
Analyze auxilary data
Build header structure
Program Options
Embed auxilary data
Extract auxilary data
Insert/Remove key
ReadBMP()
Calculate histogram for host bitmap
Process histogram selecting pair values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

Fig. 1/8

Computer code for determining pair values.

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

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

Fig. 2/8

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

Fig. 2/8

Computer code to eliminate duplicate pairs.

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

Fig. 3

Computer code constraining pair values for Truecolor images.

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

Fig. 4/8

SUBSTITUTE SHEET (RULE 26)

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

Fig. 4/8

Computer code for embedding auxiliary data.

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

Fig. 5/8

SUBSTITUTE SHEET (RULE 26)

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

Fig. 5/8

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

Fig. 5/8

SUBSTITUTE SHEET (RULE 26)

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

Fig. 5/8

Computer code to analyze lengths of runs.

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

Fig. 6/8

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

Fig. 6/8

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

return(runs);
}
```

Fig. 6/8

Computer code to set packet-start pixel flag.

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

Fig. 7/8

Computer code to embed data in two-color images.

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

Fig. 8/8

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


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

Fig. 8/8

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

Fig. 8/8

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

Fig. 8/8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US96/02357

A. CLASSIFICATION OF SUBJECT MATTER

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

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

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

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

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

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

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

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

Date of the actual completion of the international search 19 JUNE 1996	Date of mailing of the international search report 12 JUL 1996
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 PAUL R. LINTZ Telephone No. (703) 305-9600

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

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US96/02357

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

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

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

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : **G06K**	**A2**	**(11) International Publication Number:** **WO 99/36876** **(43) International Publication Date:** 22 July 1999 (22.07.99)

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

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

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

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

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

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

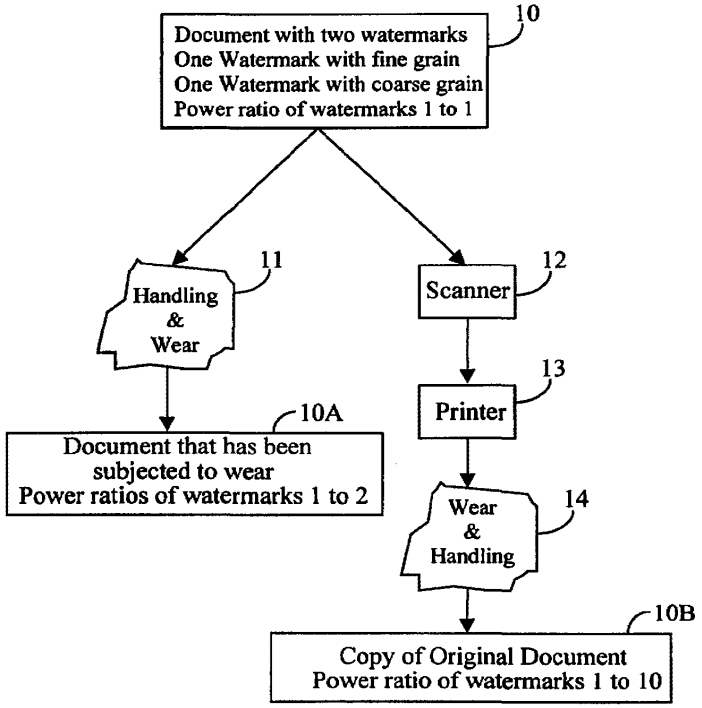
(81) Designated States: AU, BR, CA, IL, JP, KR, MX, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

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

(54) Title: MULTIPLE WATERMARKING TECHNIQUES

(57) Abstract

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



FOR THE PURPOSES OF INFORMATION ONLY

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

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

MULTIPLE WATERMARKING TECHNIQUES

Field of the Invention:

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

Background of the Invention:

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

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

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

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

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

Summary of the Invention:

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

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

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

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

Brief Description of the Figures:

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

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

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

Figure 4 illustrates a fourth embodiment of the invention.

Detailed Description of preferred embodiments:

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

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

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

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

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

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

are:

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

2) a RGB to CMYK conversion,

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

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

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

In the first embodiment:

First watermark: Has fine grain

Second watermark: Has a course grain

In the second embodiment:

First watermark: Has geometrically linear assignment of pixels

Second watermark: Has geometrically random assignment of pixels.

In the third embodiment:

First watermark: Has low power

Second watermark: Has higher power

In the fourth embodiment:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

First calculate:

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

m = min (R,G,B)

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

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

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

Then calculate I, S, and H as follows:

a) I = (M + M) /2

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

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

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

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

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

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

if H > or = 360 then H = H - 360

if H < 0 then H = H + 360

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

First calculate:

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

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

m = 2I - M

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

Then calculate R, G and B as follows:

a) if H < 60 then R = M

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

if H < 240 then R = m

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

otherwise R = M

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

if H < 180 then G = M

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

otherwise G = m

c) if H < 120 then B = m

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

if H < 300 then B = M

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

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

Calculate new values for H and S as follows:

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

else H = H + (H1 -H)x

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

else S = S +(S1 -S) x

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

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

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

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

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

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

I claim

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

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

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

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

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

6) A method of differentiating copies of an original document from the original document comprising, said document containing a first digital watermark which has a first set of characteristics and a second digital watermark which has a second set of characteristics,

reading said first and second watermarks from said original document and comparing the resultant values to generate a first set of results,

reading said first and second watermarks from said copy of said original document and comparing the resulting values to generate a second set of results, and

using differences between said first and second sets of results to differentiate an original document from a copy of said original document.

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

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

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

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

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

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

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

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

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

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

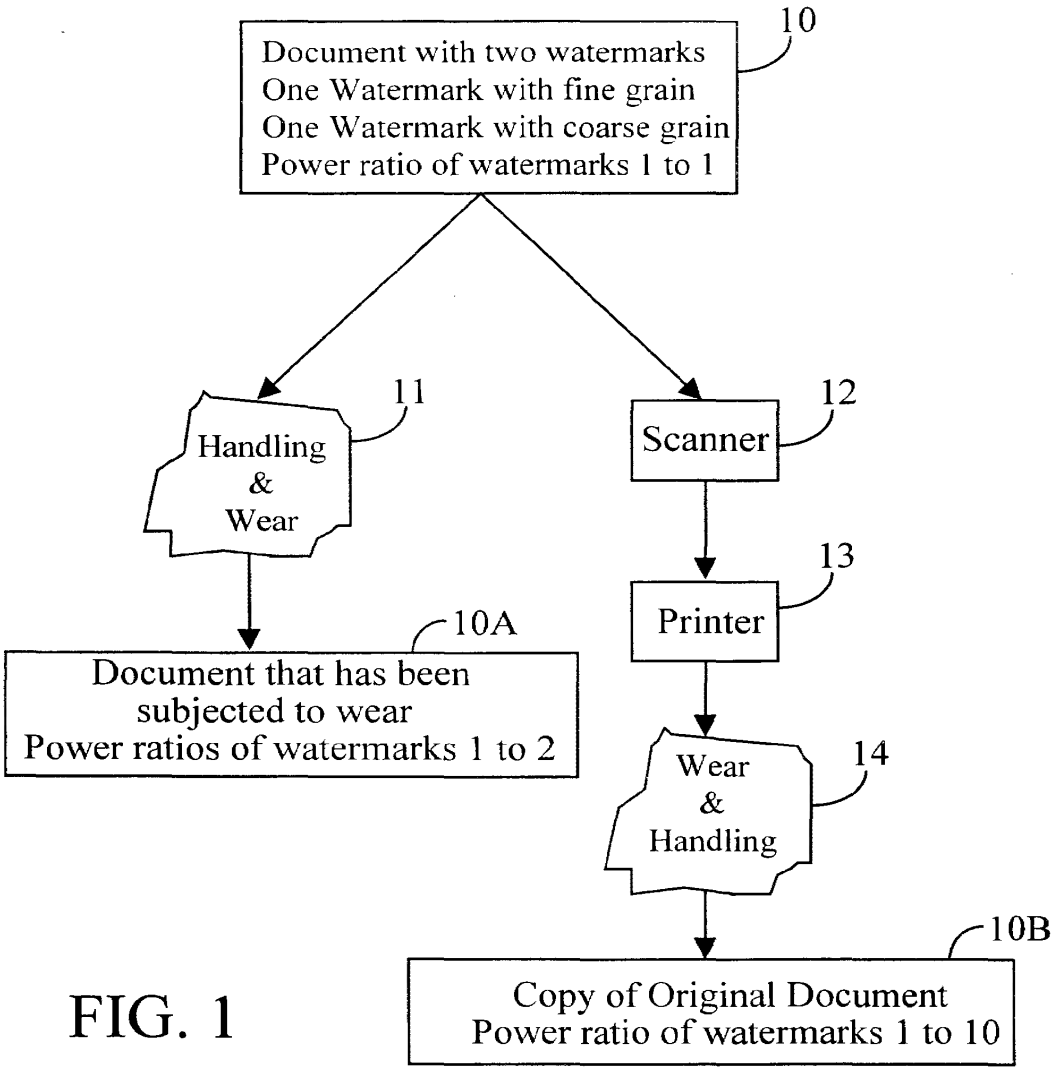

Document with two watermarks
One Watermark with fine grain
One Watermark with coarse grain
Power ratio of watermarks 1 to 1
10
Handling & Wear
11
Scanner
12
Printer
13
Wear & Handling
14
Document that has been subjected to wear
Power ratios of watermarks 1 to 2
10A
Copy of Original Document
Power ratio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

FIG. 1

FIG. 2A

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

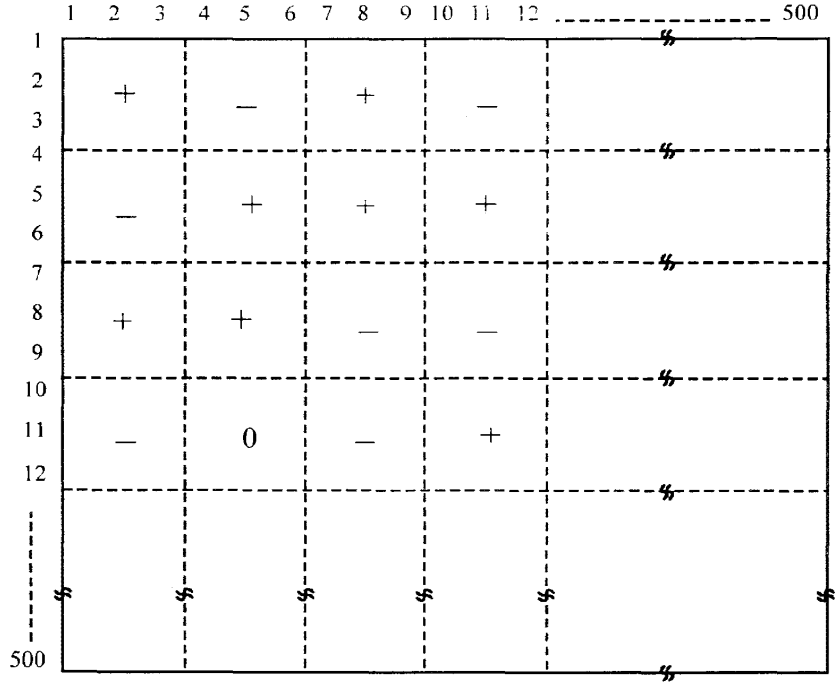


FIG. 2B

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

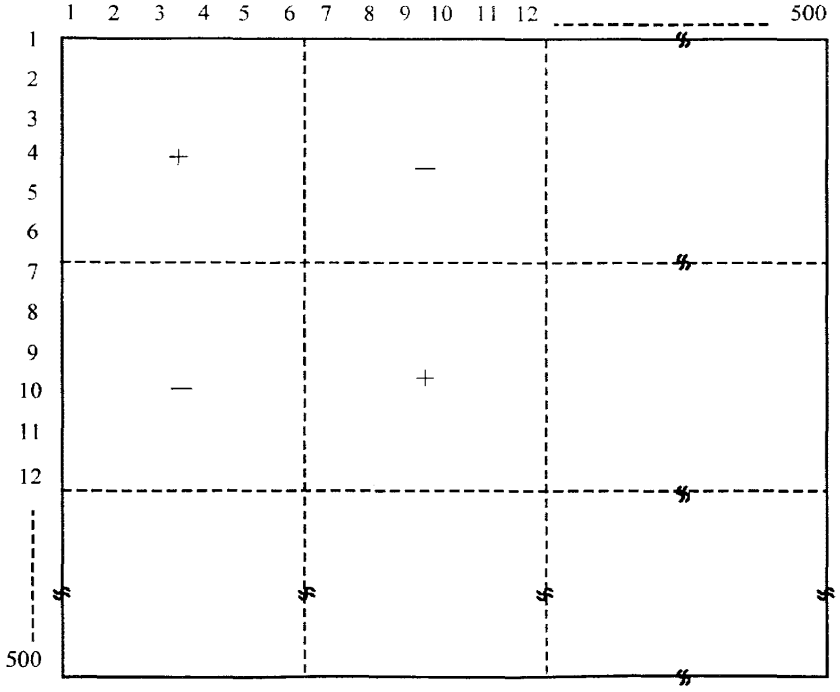


FIG. 3A Geometrically linear assignment of pixels to each bit

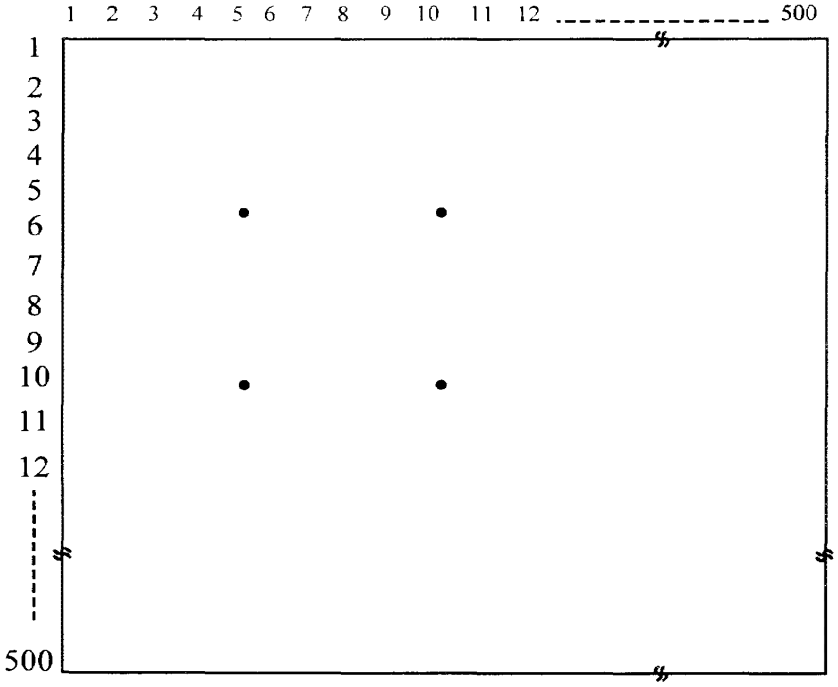
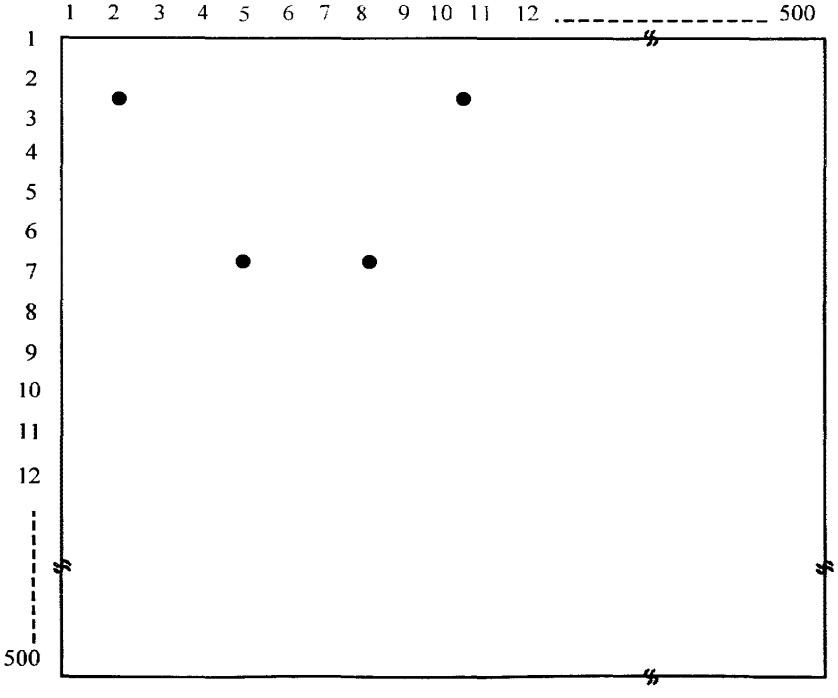


FIG. 3B Geometrically random assignment of pixels to each bit



(1) RGB ⟶ HSI

(2) First Watermark

$$HSI + WM1_{\Delta} \xrightarrow{T} RGB1$$

(3) Second Watermark

$$HSI + \text{Biased } WM2_{\Delta} \longrightarrow RGB2$$

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

FIG. 4

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

(19) World Intellectual Property Organization
International Bureau

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

PCT

(10) International Publication Number
WO 01/15021 A2

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

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

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

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

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

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

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

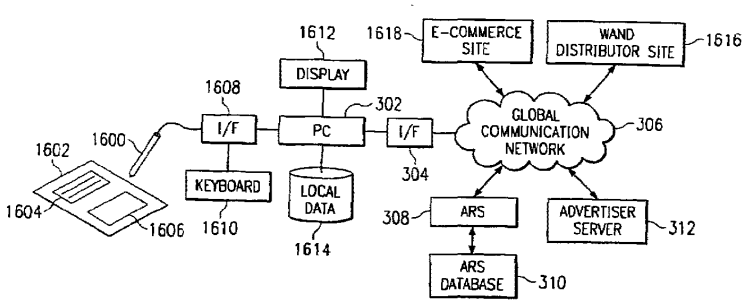
(74) Agents: HOWISON, Gregory, M. et al.; Howison, Chauza, Handley & Arnott, L.L.P., P.O. Box 741715, Dallas, TX 75374-1715 (US).

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

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

[Continued on next page]

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



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

WO 01/15021 A2

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

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

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

TECHNICAL FIELD OF THE INVENTION

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

CROSS-REFERENCE TO RELATED APPLICATIONS

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

BACKGROUND OF THE INVENTION

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

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

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

SUMMARY OF THE INVENTION

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

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

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

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

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

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

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

BRIEF DESCRIPTION OF THE DRAWINGS

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

FIGURE 1 illustrates a block diagram of the preferred embodiment;

FIGURE 2 illustrates the computer components employed in this embodiment;

FIGURE 3 illustrates system interactions over a global network;

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

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

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

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

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

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

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

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

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

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

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

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

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

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

FIGURE 17 illustrates the conversion circuit of the wedge interface;

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

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

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

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

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

FIGURE 23 illustrates a flowchart according to the ARS; and

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

DETAILED DESCRIPTION OF THE INVENTION

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

WHAT IS CLAIMED IS:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

means for launching said browser on said computer.

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

means for launching said browser to said predetermined destination.

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

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

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

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

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

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

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

transmitting converted audio signal information to the computer.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

launching said browser to the predetermined destination.

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

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

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

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

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

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

extracting product information contained within the bar code;

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

read the machine readable code;

decoding the product code; and

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

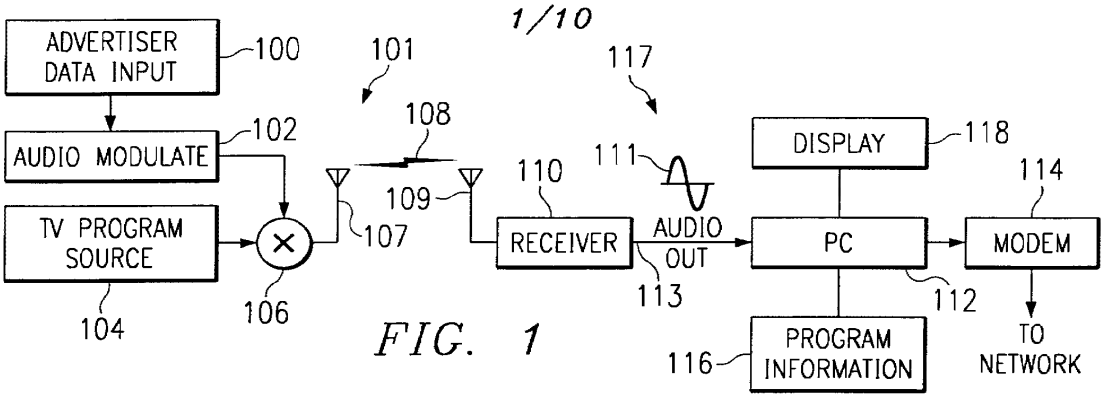


FIG. 1

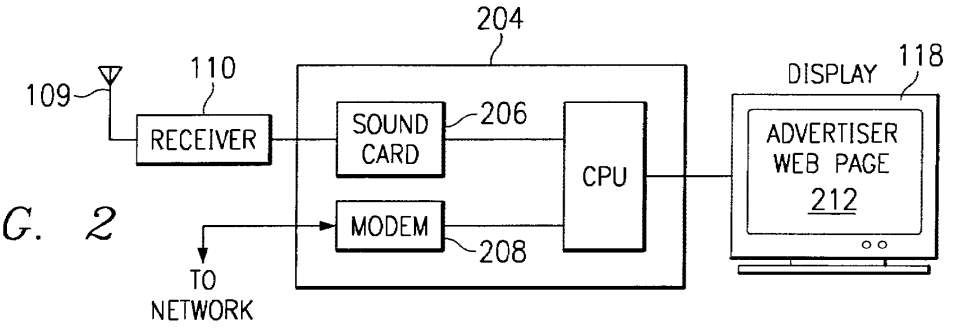


FIG. 2

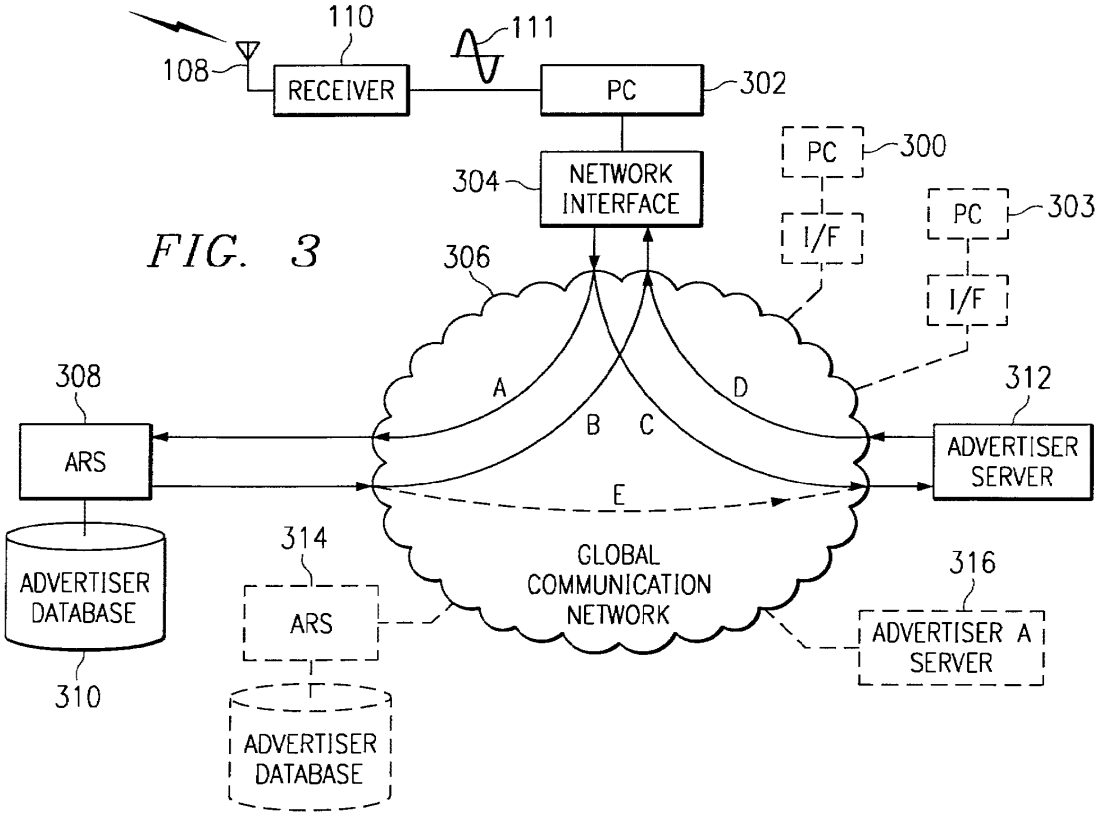


FIG. 3

PATH A: SOURCE TO ARS



FIG. 4a

PATH B: ARS TO SOURCE



FIG. 4b

PATH C: SOURCE TO ADVERTISER



FIG. 4c

PATH D: ADVERTISER TO SOURCE

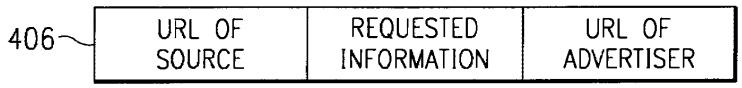


FIG. 4d

PATH E: ARS TO ADVERTISER (OPTIONAL)



FIG. 4e

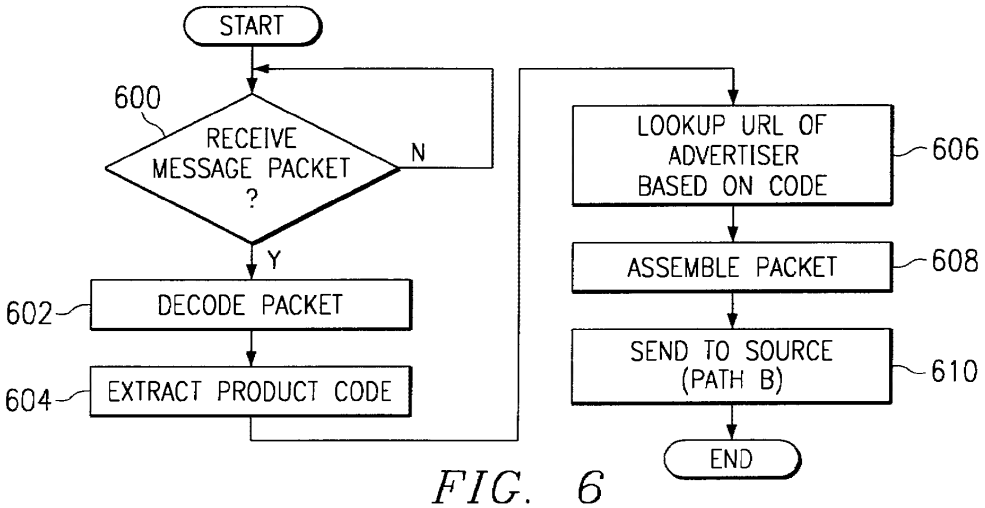


FIG. 6

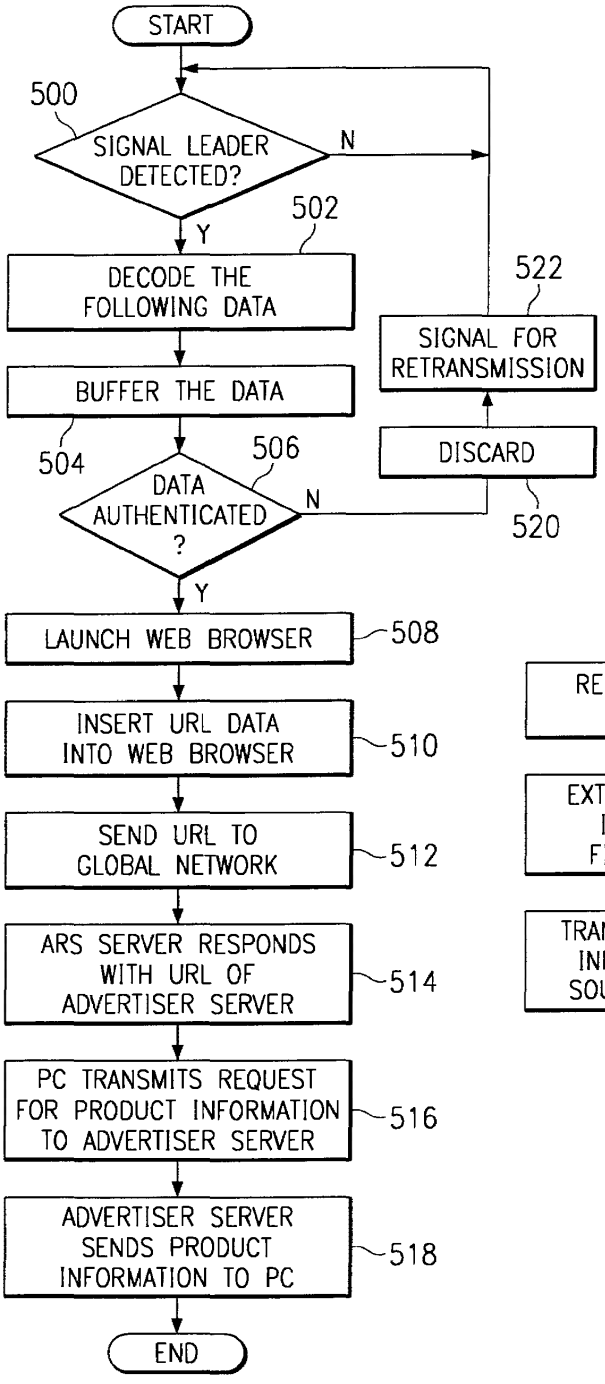

START
500
SIGNAL LEADER DETECTED?
N
Y
502
DECODE THE FOLLOWING DATA
BUFFER THE DATA
504
506
DATA AUTHENTICATED ?
N
Y
522
SIGNAL FOR RETRANSMISSION
DISCARD
520
LAUNCH WEB BROWSER
508
INSERT URL DATA INTO WEB BROWSER
510
SEND URL TO GLOBAL NETWORK
512
ARS SERVER RESPONDS WITH URL OF ADVERTISER SERVER
514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

FIG. 5

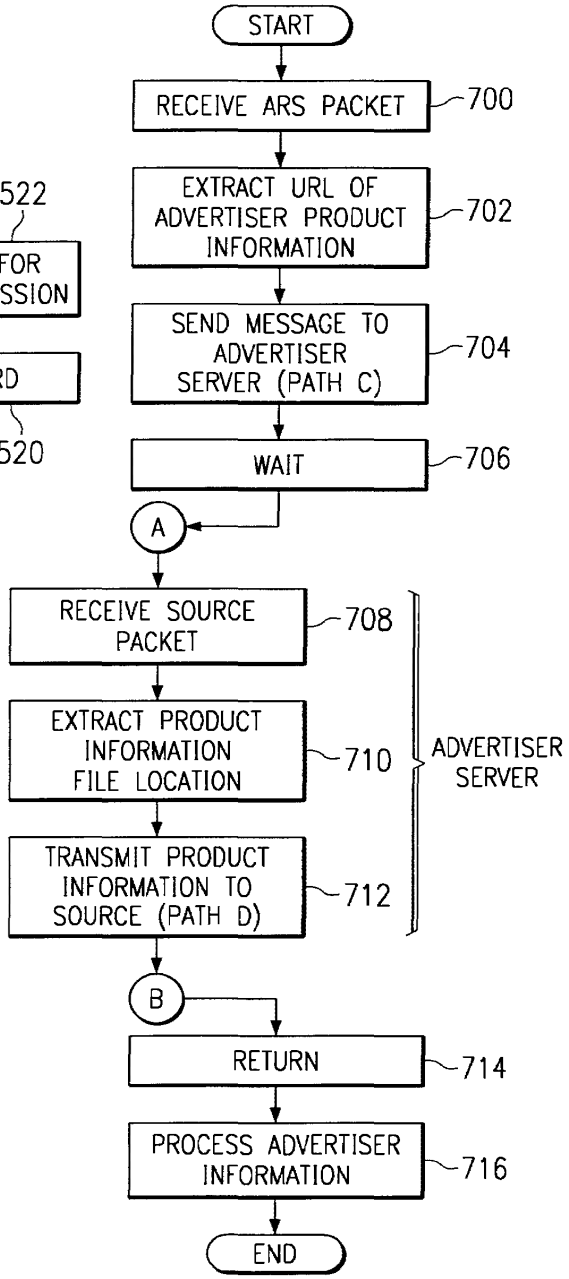

START
RECEIVE ARS PACKET
700
EXTRACT URL OF ADVERTISER PRODUCT INFORMATION
702
SEND MESSAGE TO ADVERTISER SERVER (PATH C)
704
WAIT
706
A
RECEIVE SOURCE PACKET
708
EXTRACT PRODUCT INFORMATION FILE LOCATION
710
TRANSMIT PRODUCT INFORMATION TO SOURCE (PATH D)
712
ADVERTISER SERVER
B
RETURN
714
PROCESS ADVERTISER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

FIG. 7

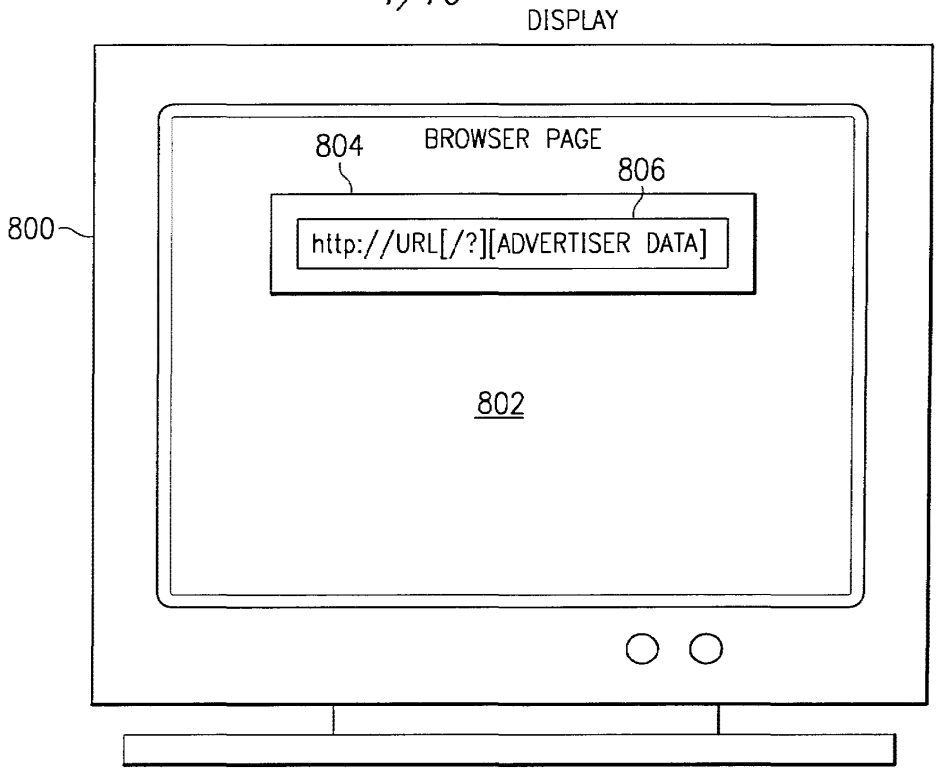
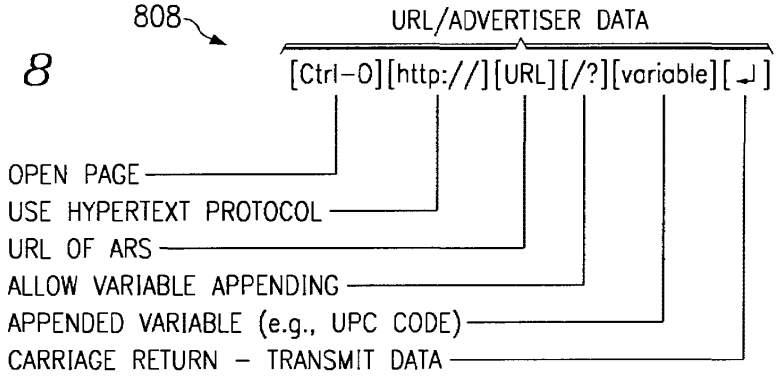

DISPLAY
800
804
BROWSER PAGE
806
http://URL[/?][ADVERTISER DATA]
802

FIG. 8


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

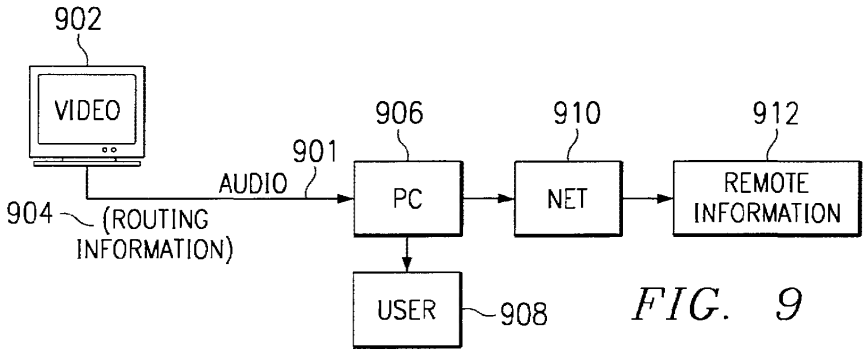

902
VIDEO
906
910
912
901
AUDIO
PC
NET
REMOTE INFORMATION
904
(ROUTING INFORMATION)
USER
908

FIG. 9

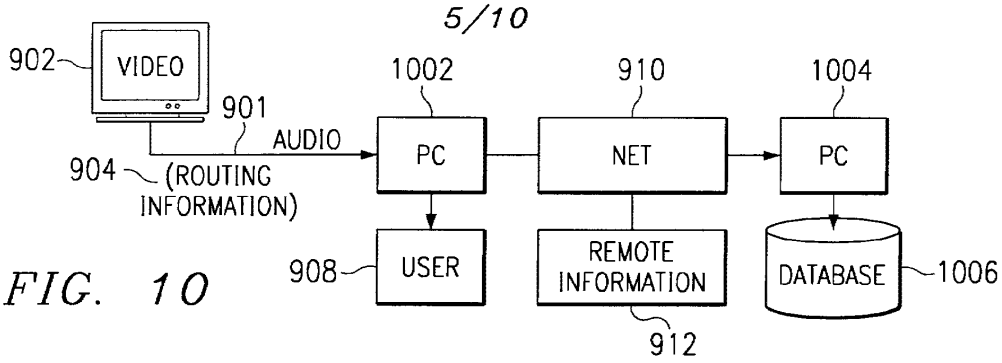


FIG. 10

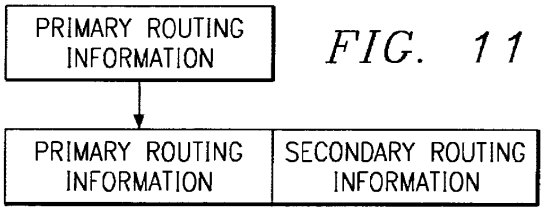


FIG. 11

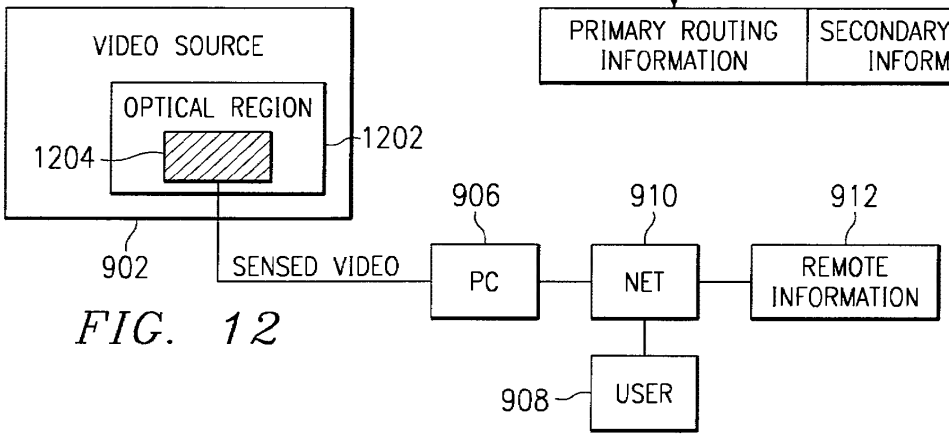


FIG. 12

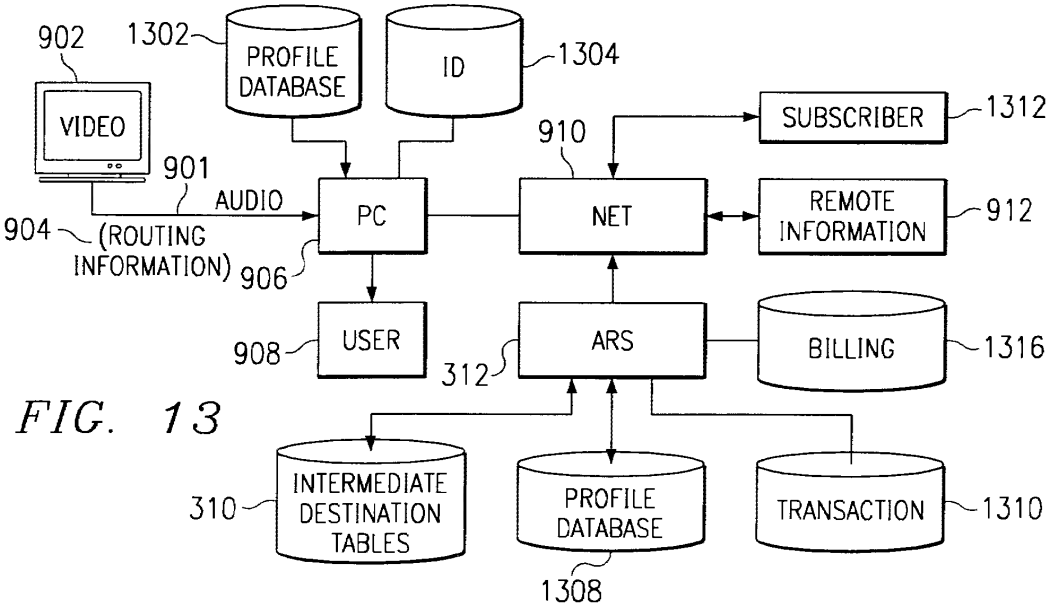


FIG. 13

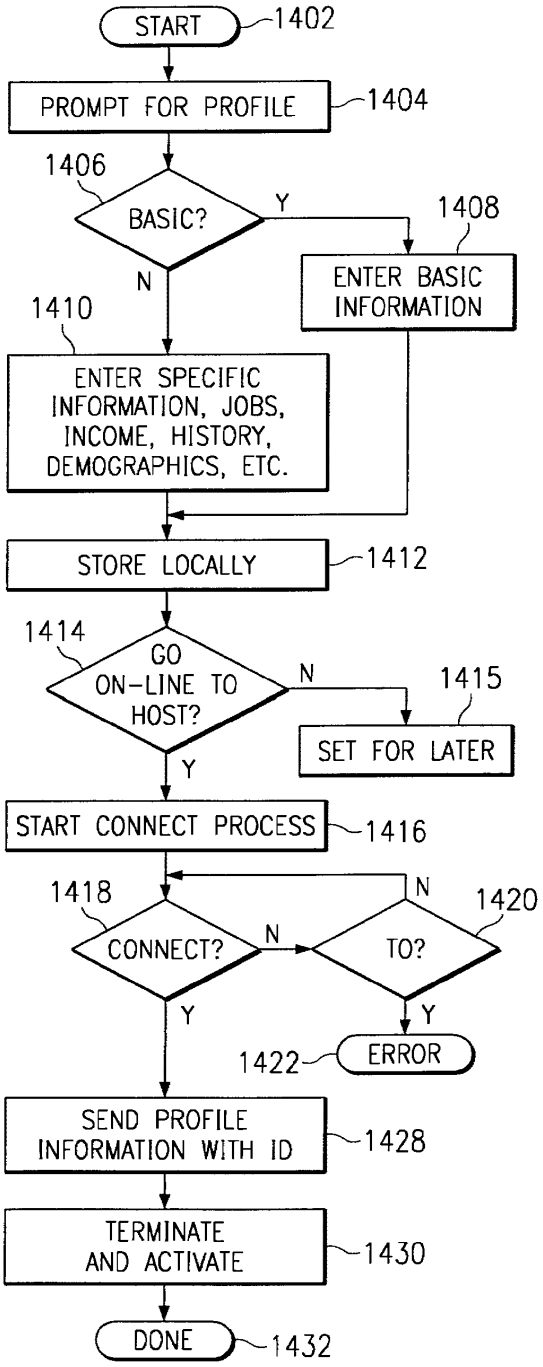

START
1402
PROMPT FOR PROFILE
1404
1406
BASIC?
Y
1408
ENTER BASIC INFORMATION
N
1410
ENTER SPECIFIC INFORMATION, JOBS, INCOME, HISTORY, DEMOGRAPHICS, ETC.
STORE LOCALLY
1412
1414
GO ON-LINE TO HOST?
N
1415
SET FOR LATER
Y
START CONNECT PROCESS
1416
1418
CONNECT?
N
TO?
1420
N
Y
Y
1422
ERROR
SEND PROFILE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

FIG. 14

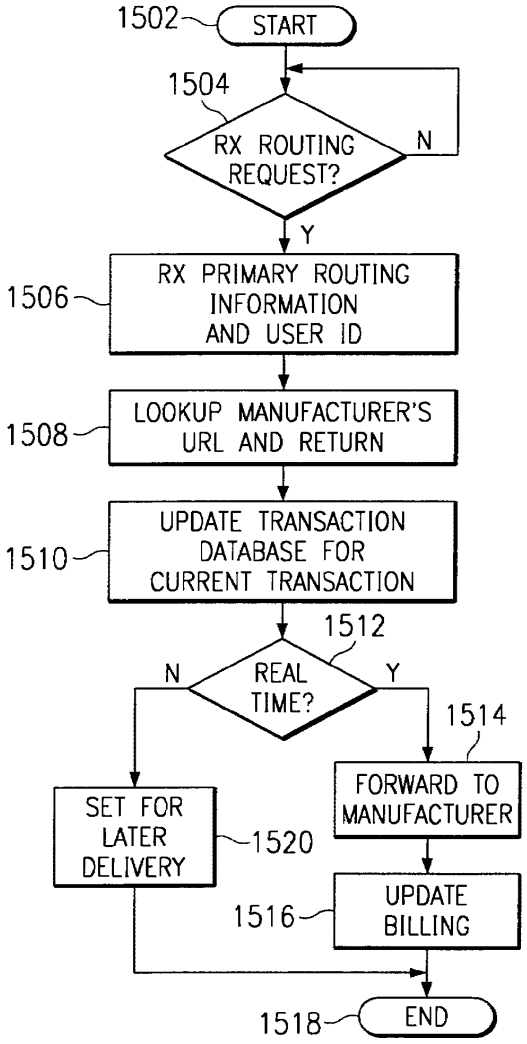

1502
START
1504
RX ROUTING REQUEST?
N
Y
1506
RX PRIMARY ROUTING INFORMATION AND USER ID
1508
LOOKUP MANUFACTURER'S URL AND RETURN
1510
UPDATE TRANSACTION DATABASE FOR CURRENT TRANSACTION
1512
N
REAL TIME?
Y
1514
SET FOR LATER DELIVERY
1520
FORWARD TO MANUFACTURER
1516
UPDATE BILLING
1518
END

FIG. 15

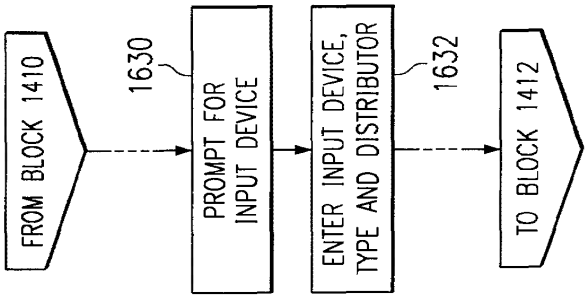


FIG. 16a

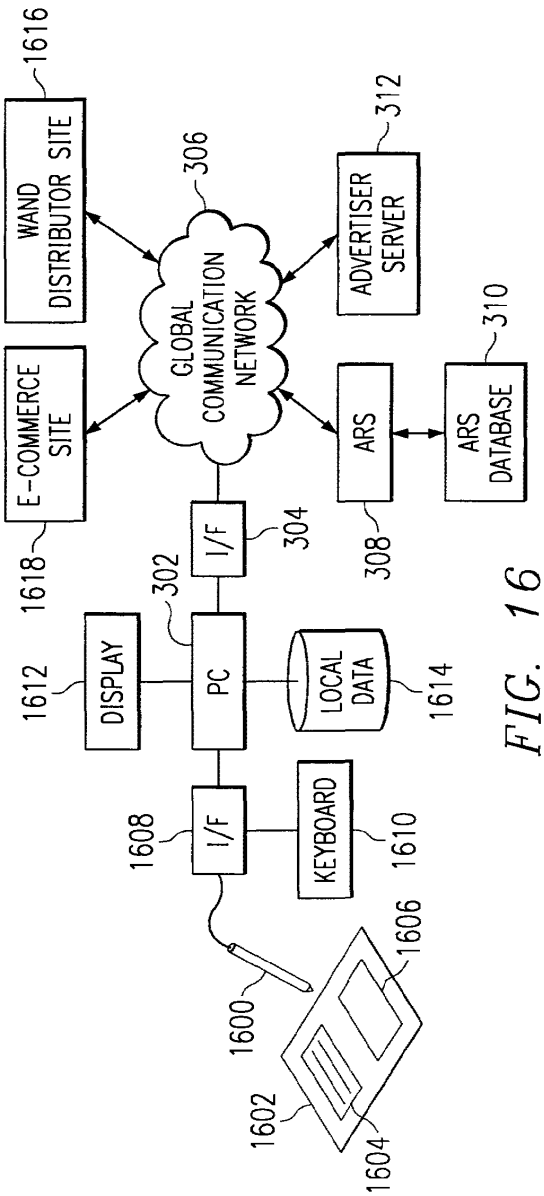


FIG. 16

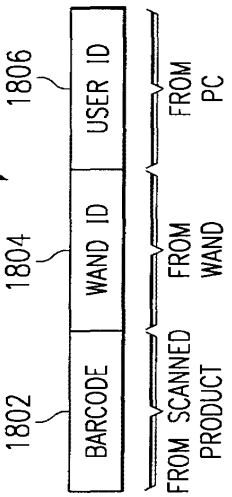


FIG. 18

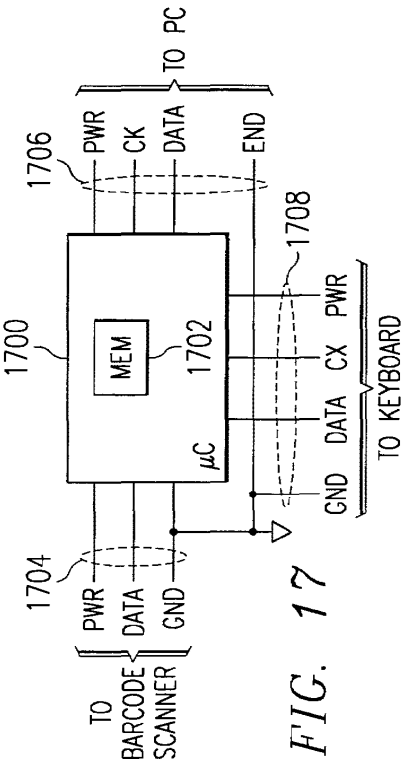
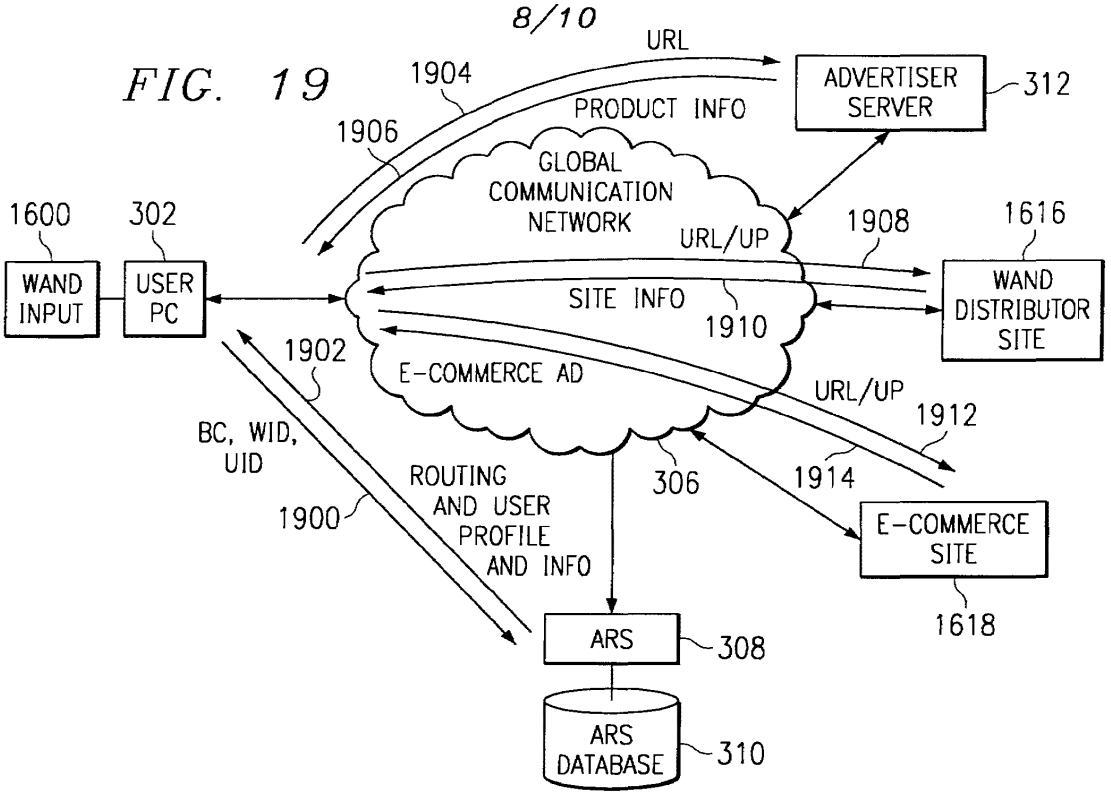


FIG. 17

FIG. 19



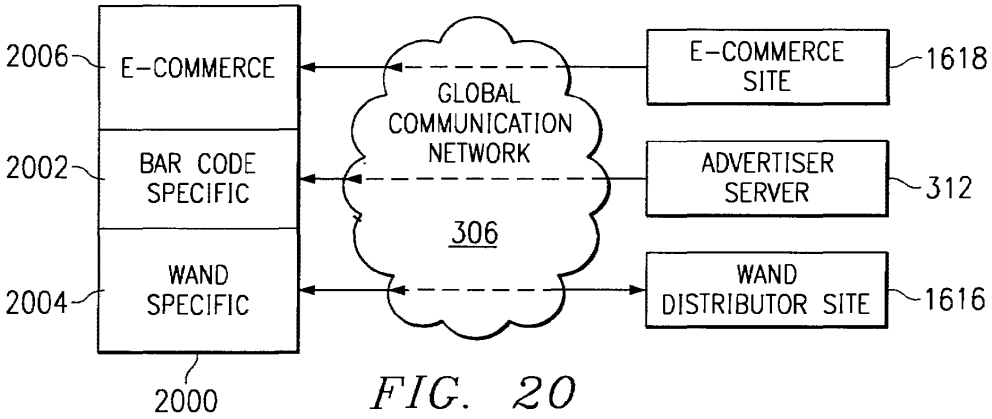


FIG. 20

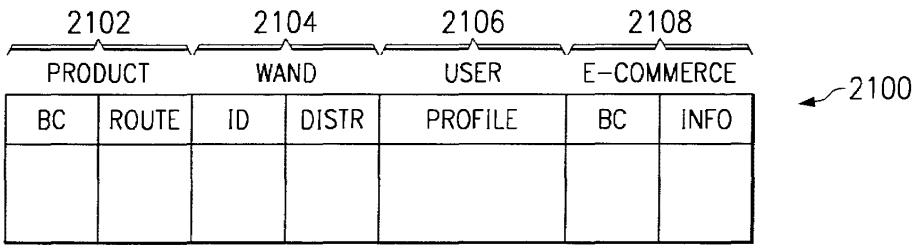


FIG. 21

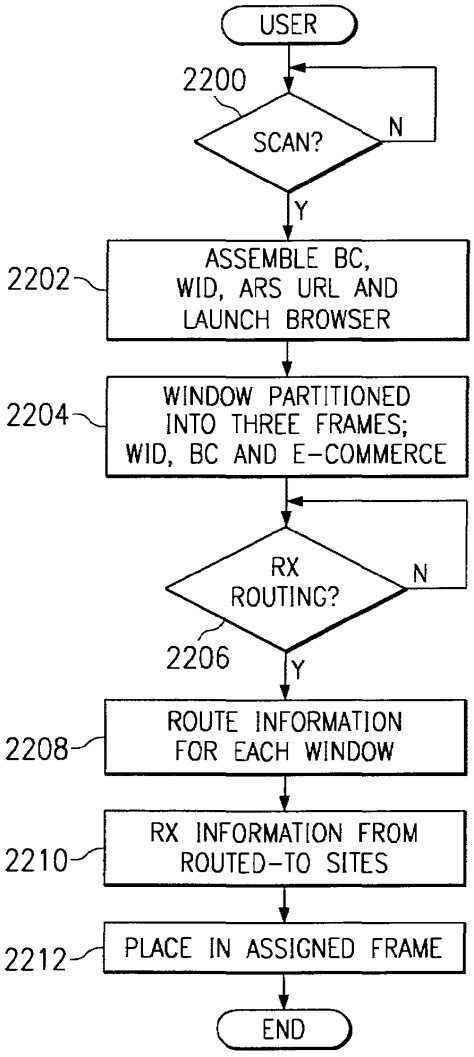

USER
2200
SCAN?
N
Y
2202
ASSEMBLE BC, WID, ARS URL AND LAUNCH BROWSER
2204
WINDOW PARTITIONED INTO THREE FRAMES; WID, BC AND E-COMMERCE
RX ROUTING?
N
2206
Y
2208
ROUTE INFORMATION FOR EACH WINDOW
2210
RX INFORMATION FROM ROUTED-TO SITES
2212
PLACE IN ASSIGNED FRAME
END

FIG. 22

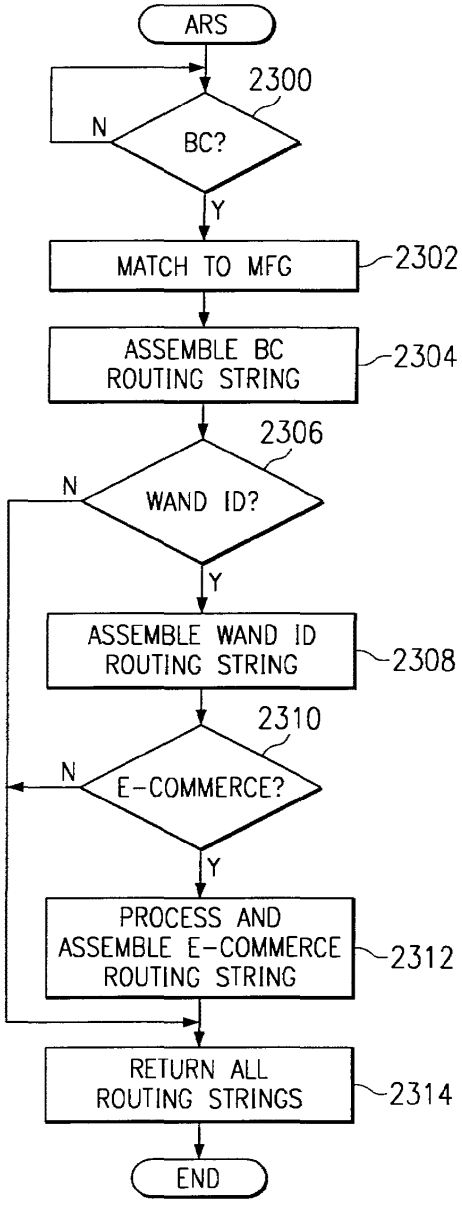

ARS
2300
N
BC?
Y
MATCH TO MFG
2302
ASSEMBLE BC ROUTING STRING
2304
2306
N
WAND ID?
Y
ASSEMBLE WAND ID ROUTING STRING
2308
2310
N
E-COMMERCE?
Y
PROCESS AND ASSEMBLE E-COMMERCE ROUTING STRING
2312
RETURN ALL ROUTING STRINGS
2314
END

FIG. 23

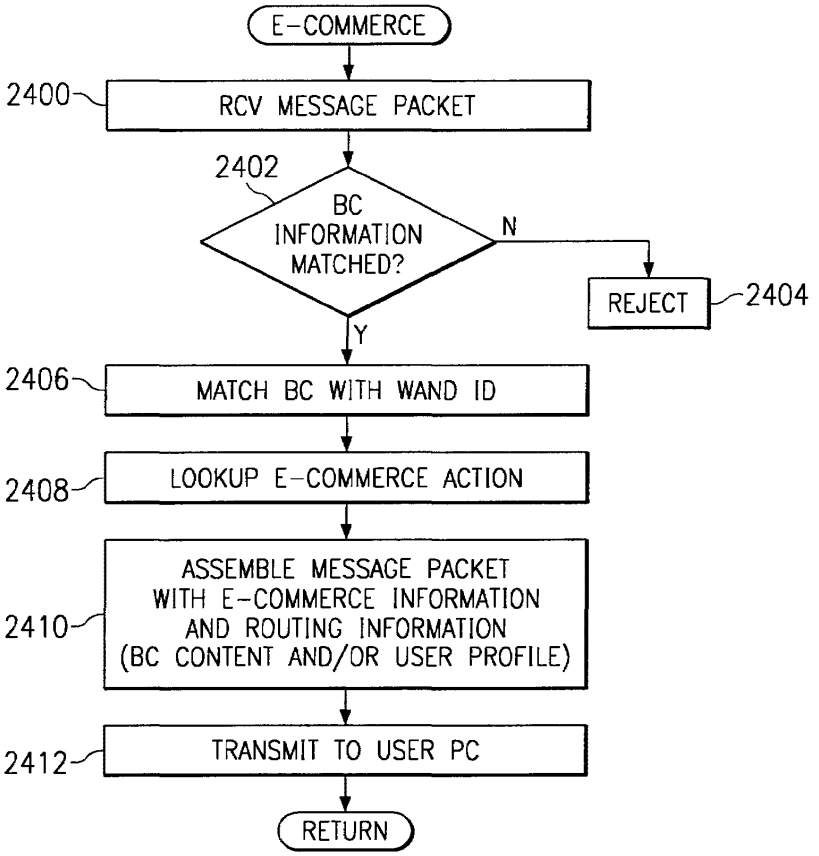

E-COMMERCE
2400
RCV MESSAGE PACKET
2402
BC INFORMATION MATCHED?
N
REJECT
2404
Y
2406
MATCH BC WITH WAND ID
2408
LOOKUP E-COMMERCE ACTION
2410
ASSEMBLE MESSAGE PACKET WITH E-COMMERCE INFORMATION AND ROUTING INFORMATION (BC CONTENT AND/OR USER PROFILE)
2412
TRANSMIT TO USER PC
RETURN

FIG. 24

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

(19) World Intellectual Property Organization
International Bureau

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

PCT

(10) International Publication Number
WO 01/61508 A1

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

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

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

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

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

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

(72) Inventors; and

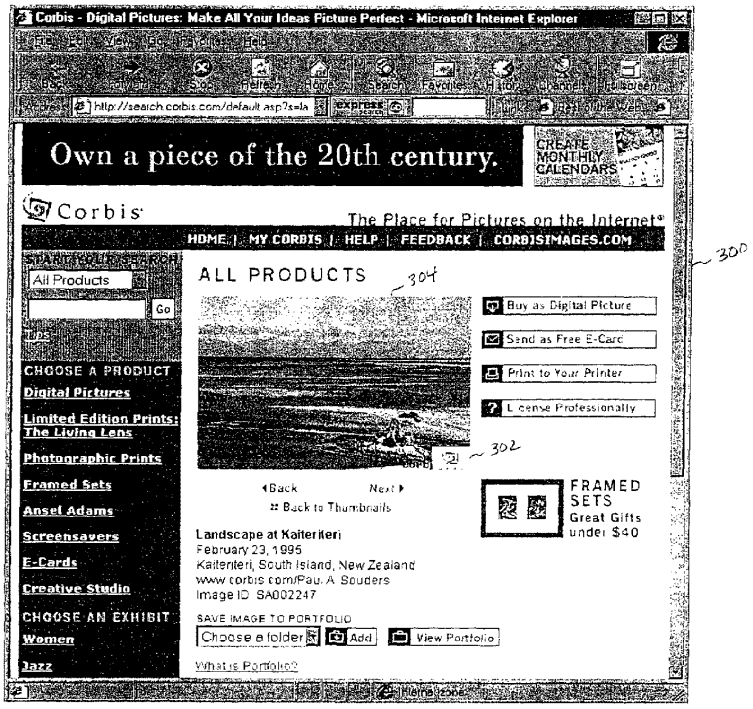
(75) Inventors/Applicants *(for US only)*: **RAMOS, Daniel, O.** [US/US]; 16869 SW Hargis Road, Beaverton, OR 97007 (US). **JONES, Kevin, C.** [US/US]; 4850 NW Neskowin Ave., Portland, OR 97229 (US). **RHOADS, Geoffrey, B.** [US/US]; 2961 SW Turner Road, West Linn, OR 97068 (US).

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

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

[Continued on next page]

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



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

WO 01/61508 A1

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

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

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

WATERMARK ENCODER AND DECODER ENABLED SOFTWARE AND DEVICES

Related Application Data

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

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

Technical Field

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

Background and Summary

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

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

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

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

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

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

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

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

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

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

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

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

Brief Description of Drawings

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

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

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

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

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

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

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

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

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

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

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

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

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

Detailed Description

Introduction

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

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

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

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

Integrating a Watermark Decoder in Operating Systems and Other Applications

In Operating System

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

In Browser

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

In Other Applications

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

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

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

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

Context Sensitive Watermark Links

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

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

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

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

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

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

Supporting Multiple Watermark Types

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

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

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

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

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

Media Object Branding

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

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

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

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

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

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

Aggregating Metadata

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

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

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

Additional Functionality

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

Metadata

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

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

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

Multiple Watermarks

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

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

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

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

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

Digital Rights Management

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

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

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

Watermark and Context Information for Usage Control

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

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

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

Asset Management

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

Integration with Directory Services

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

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

Media Object Player and Delivery Integration

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

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

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

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

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

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

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

Content Authoring Tools

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

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

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

Web Server Integration and Related Applications

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

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

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

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

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

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

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

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

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

Content Filtering and Counting

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

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

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

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

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

Watermark Based Spiders

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

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

Event Scheduling Based on Embedded Data

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

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

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

Integrating a Watermark Encoder in Operating Systems and Other Applications

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

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

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

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

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

Digital Watermarks as a Gateway and Control Mechanism

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Content Filtering and Indexing System Using Digital Watermarks

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Operating Environment for Computer Implementations

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Extensions to Other Forms of Media Object Linking

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

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

Concluding Remarks

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

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

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

We Claim:

1. A file browser system comprising:

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

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

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

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

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

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

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

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

8. A file browser system comprising:

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

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

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

10. A watermark decoder system comprising:

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

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

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

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

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

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

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

14. A method of rendering a media object comprising:

decoding an object identifier from the media object;

sending the object identifier to a metadata server;

receiving a brand identifier from the metadata server; and

displaying a representation of the brand identifier.

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

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

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

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

19. A method for extending a user interface of a media player comprising:

in response to input requesting playback of a media object, extracting an object identifier from the media object;

using the object identifier to look up metadata associated with the media object;

extending a user interface of a media player to include a representation of the metadata associated with the media object.

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

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

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

22. A system which includes an e-mail server connected to the Internet

means for transmitting messages from individual user to said e-mail server,

watermark detecting means for detecting and reading watermarks in e-mail messages before such messages are transmitted from said e-mail server to the Internet,

means for preventing the transmission of messages from said e-mail server to the Internet if said watermark detecting means detects a watermark which has an indication that the message including said watermark is confidential.

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

24. A method of controlling the distribution of electronic messages that include confidential information, said messages including digital watermarks which carry data indicating that the message includes confidential information,

reading watermarks in messages prior to transmission of said messages, and

controlling the distribution of each electronic message which includes a watermark in response to the data carried by the watermark in the message.

25. A method of transmitting electronic messages from a sender to a receiver which comprises,

detecting and reading digital watermarks embedded in such messages to determine how the flags in such watermarks are set,

interrogating a database to determine what action should be taken with a message based upon the identity of the sender, the identity of the receiver and the flag settings in the watermark in the message.

26. A content filtering system in which watermark content filters are distributed in programs executing on computers interconnected via a communications network, the watermark content filters being operable to detect digital watermarks in content entering a location in the network, to read watermark payload information, and to take action to control rendering or transfer of the content based on content type flags in the watermark payload information.

27. The system of claim 26 wherein the content filters report detection events to a searchable database enabling users to search for content and find a network address indicating where the content has been detected by the content filters.

28. The system of claim 26 wherein the content filters execute rules that control rendering or transfer of content based on a user ID and content type flags in the digital watermarks.

29. The system of claim 26 wherein content type flags are used in combination with rules relating to a user's access rights to enable or disable rendering of advertising along with rendering of the content.

30. A distributed watermark spider system comprising:

digital watermark detectors implemented in spider thread programs executing in devices distributed on a communications network; the spider thread programs operable to screen content at corresponding devices for watermarks, and when found, report information from watermark payloads encoded into the watermark and information about the event to a spider program executing on device connected to the network.

31. The system of claim 30 wherein the information about the event includes a network address of detected content enabling remote devices to access the detected content.

32. The system of claim 31 wherein the spider program is in communication with one or more searchable databases storing the watermark payload information, information about the events, and information about the content cross referenced to the content, wherein the searchable database enables users to search for content by name and find related information about the content, including an address where the information can be fetched via the system.

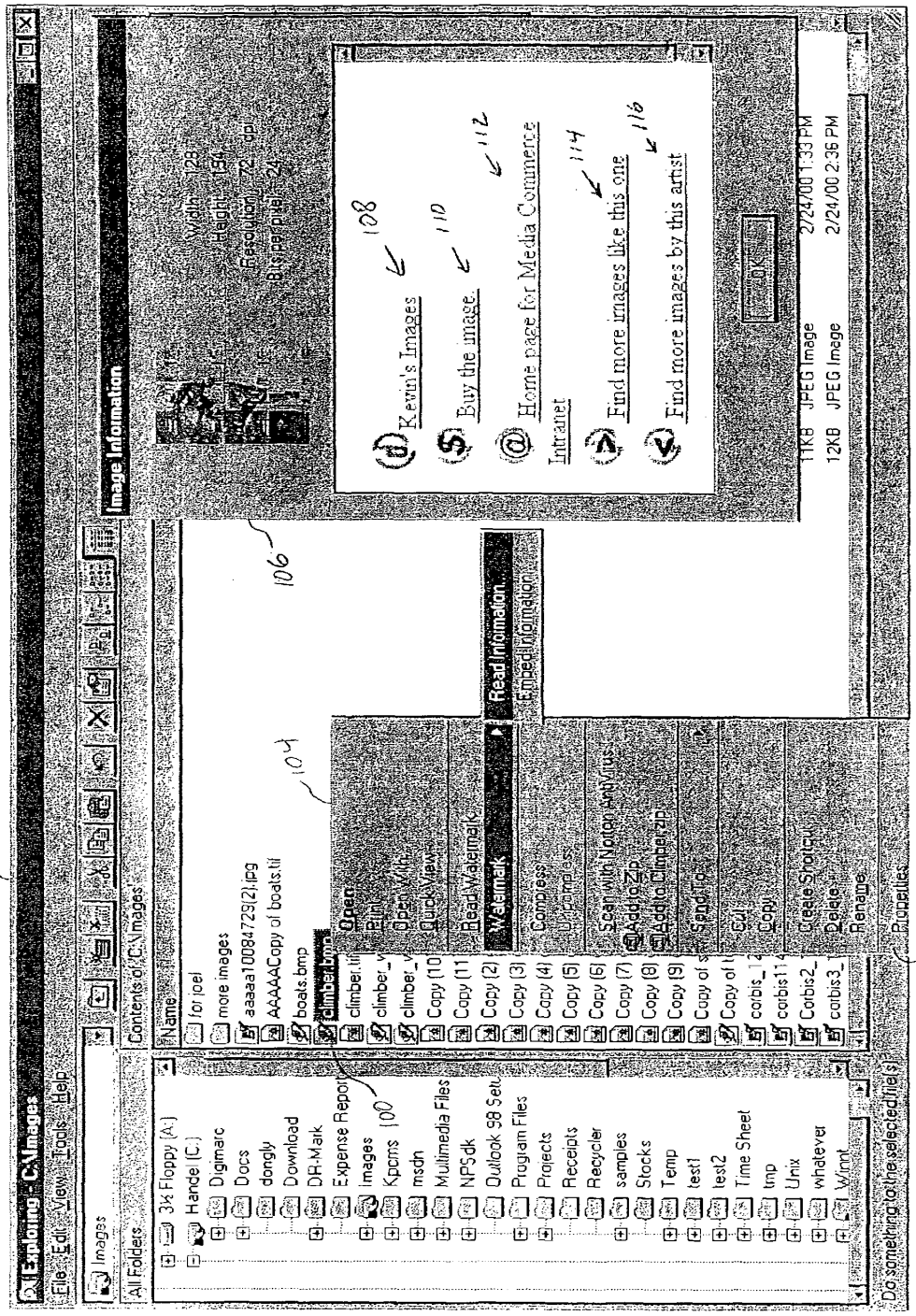

Exploring - C:\Images
File Edit View Tools Help
Images
All Folders
Contents of 'C:\Images'
Name
Image Information
Width: 128
Height: 154
Resolution: 72 dpi
BitsPerPixel: 24
Kevin's Images
Buy the image
Home page for Media Commerce Intranet
Find more images like this one
Find more images by this artist
Read Information...
Embed Information...
Watermark...
Read Watermark
Open
Print...
Open With...
Quick View
Compress
Uncompress
Scan with Norton AntiVirus
Add to Zip
Add to climber.zip
Send To
Cut
Copy
Create Shortcut
Delete
Rename
Properties
OK
11KB JPEG Image 2/24/00 1:33 PM
12KB JPEG Image 2/24/00 2:36 PM
Do something to the selected file(s)
for joel
more images
aaaaa10084729(2).jpg
AAAAACopy of boats.tif
boats.bmp
climber.bmp
Digimarc
Docs
dongly
Download
DR-Mark
Expense Report
images
Kpcms
msdn
Multimedia Files
NPSdk
Outlook 98 Sec
Program Files
Projects
Receipts
Recycler
samples
Stocks
Temp
test1
test2
Time Sheet
tmp
Unix
whatever
Wmnt
102
100
104
106
108
110
112
114
116
118

Fig. 1

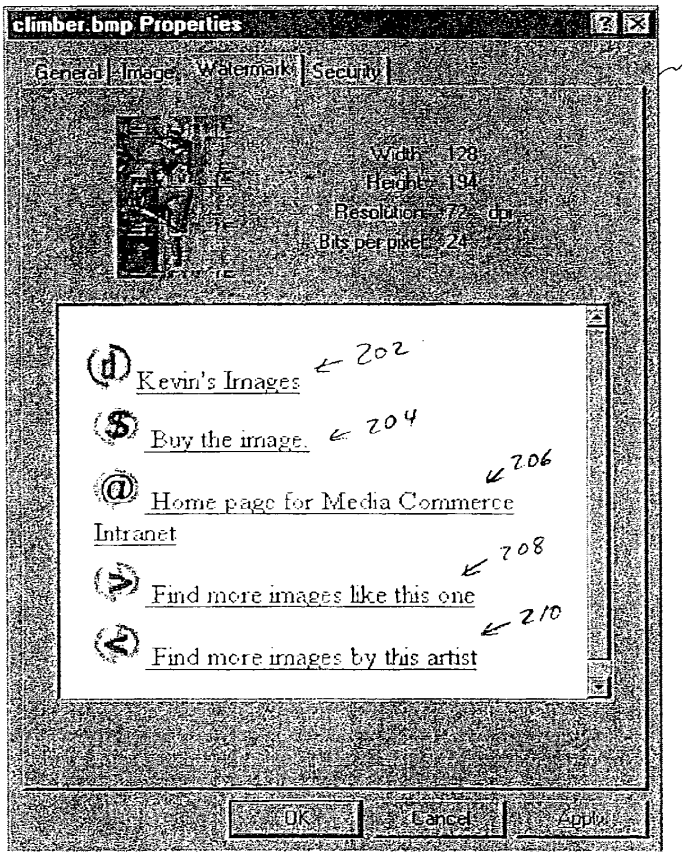

climber.bmp Properties
General
Image
Watermark
Security
Width: 128
Height: 194
Resolution: 72 dpi
Bits per pixel: 24
Kevin's Images
Buy the image.
Home page for Media Commerce Intranet
Find more images like this one
Find more images by this artist
OK
Cancel
Apply
200
202
204
206
208
210

Fig. 2

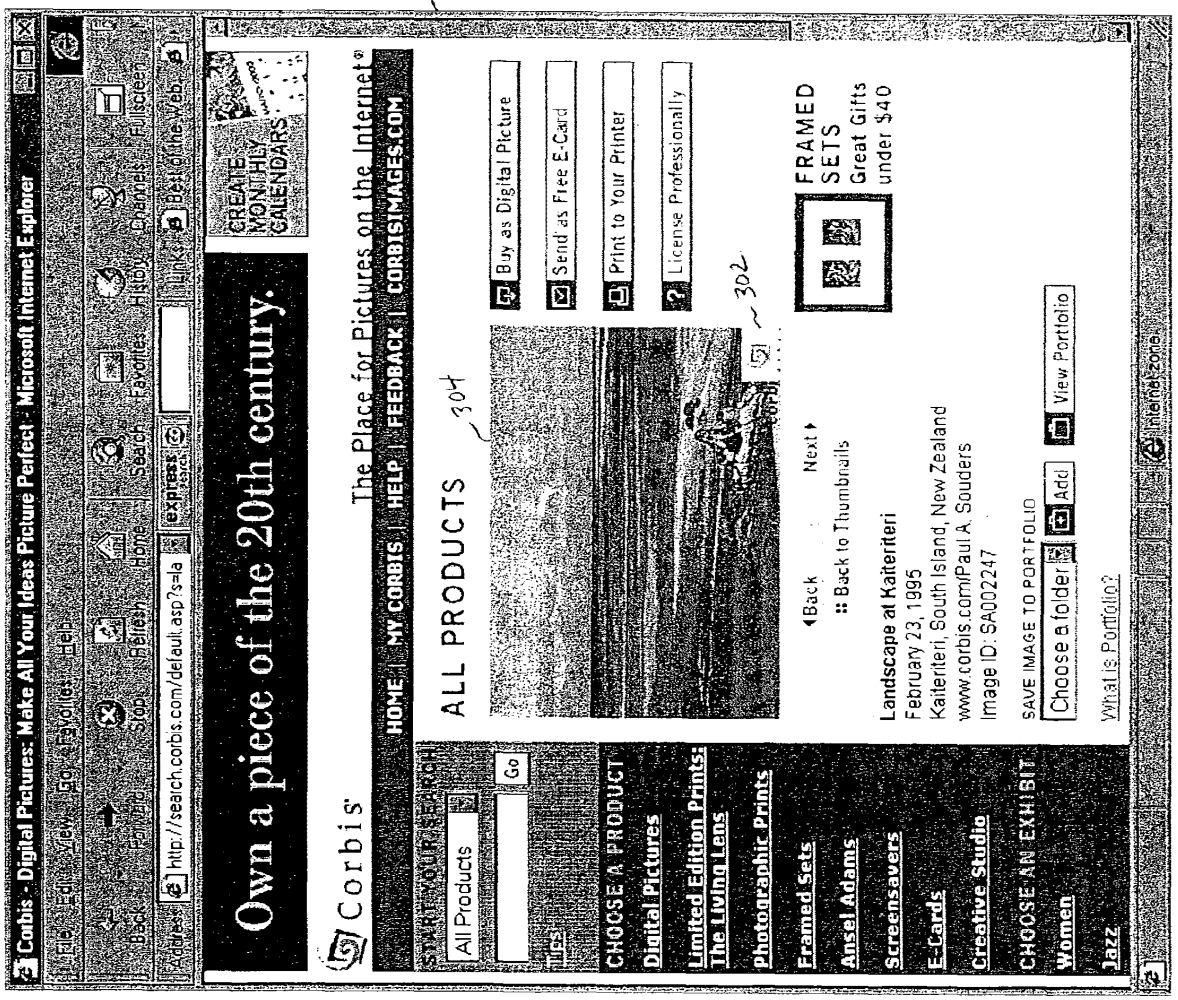

Corbis - Digital Pictures: Make All Your Ideas Picture Perfect - Microsoft Internet Explorer
Own a piece of the 20th century.
Corbis
The Place for Pictures on the Internet®
HOME | MY CORBIS | HELP | FEEDBACK | CORBISIMAGES.COM
CREATE MONTHLY CALENDARS
START YOUR SEARCH
All Products
Go
Tips
CHOOSE A PRODUCT
Digital Pictures
Limited Edition Prints: The Living Lens
Photographic Prints
Framed Sets
Ansel Adams
Screensavers
E-Cards
Creative Studio
CHOOSE AN EXHIBIT
Women
Jazz
ALL PRODUCTS
Buy as Digital Picture
Send as Free E-Card
Print to Your Printer
License Professionally
Back
Next
Back to Thumbnails
FRAMED SETS
Great Gifts under $40
Landscape at Kaiteriteri
February 23, 1995
Kaiteriteri, South Island, New Zealand
www.corbis.com/Paul A. Souders
Image ID: SA002247
SAVE IMAGE TO PORTFOLIO
Choose a folder
Add
View Portfolio
What is Portfolio?
300
304
302

Fig. 3

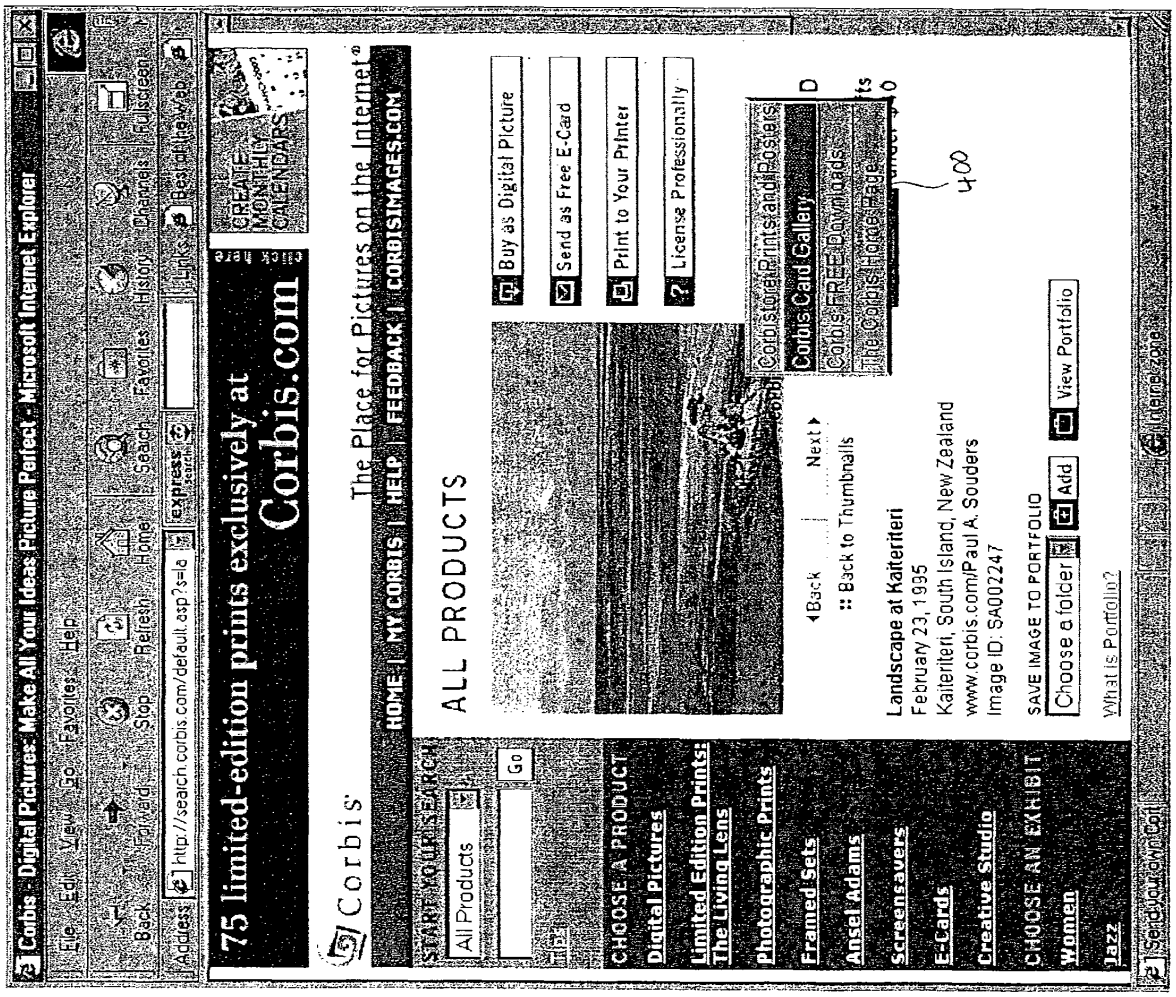

Corbis - Digital Pictures: Make All Your Ideas Picture Perfect - Microsoft Internet Explorer
75 limited-edition prints exclusively at Corbis.com
The Place for Pictures on the Internet
HOME | MY CORBIS | HELP | FEEDBACK | CORBISIMAGES.COM
ALL PRODUCTS
Buy as Digital Picture
Send as Free E-Card
Print to Your Printer
License Professionally
CHOOSE A PRODUCT
Digital Pictures
Limited Edition Prints: The Living Lens
Photographic Prints
Framed Sets
Ansel Adams
Screensavers
E-Cards
Creative Studio
CHOOSE AN EXHIBIT
Women
Jazz
Corbis Card Gallery
Landscape at Kaiteriteri
February 23, 1995
Kaiteriteri, South Island, New Zealand
www.corbis.com/Paul A. Souders
Image ID: SA002247
SAVE IMAGE TO PORTFOLIO
Choose a folder
Add
View Portfolio
What is Portfolio?
400

Fig. 4

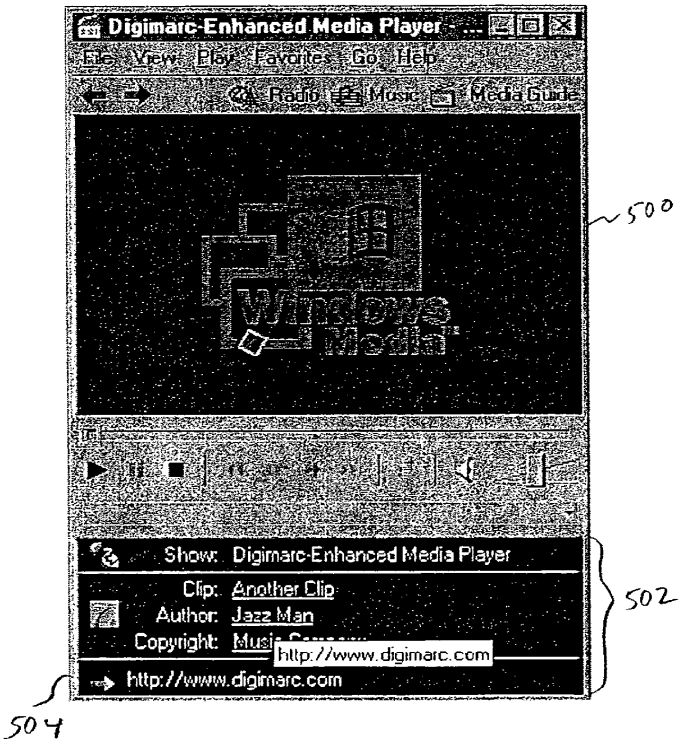

Digimarc-Enhanced Media Player
File View Play Favorites Go Help
Radio Music Media Guide
Windows Media
Show: Digimarc-Enhanced Media Player
Clip: Another Clip
Author: Jazz Man
Copyright:
http://www.digimarc.com
http://www.digimarc.com
500
502
504

Fig. 5

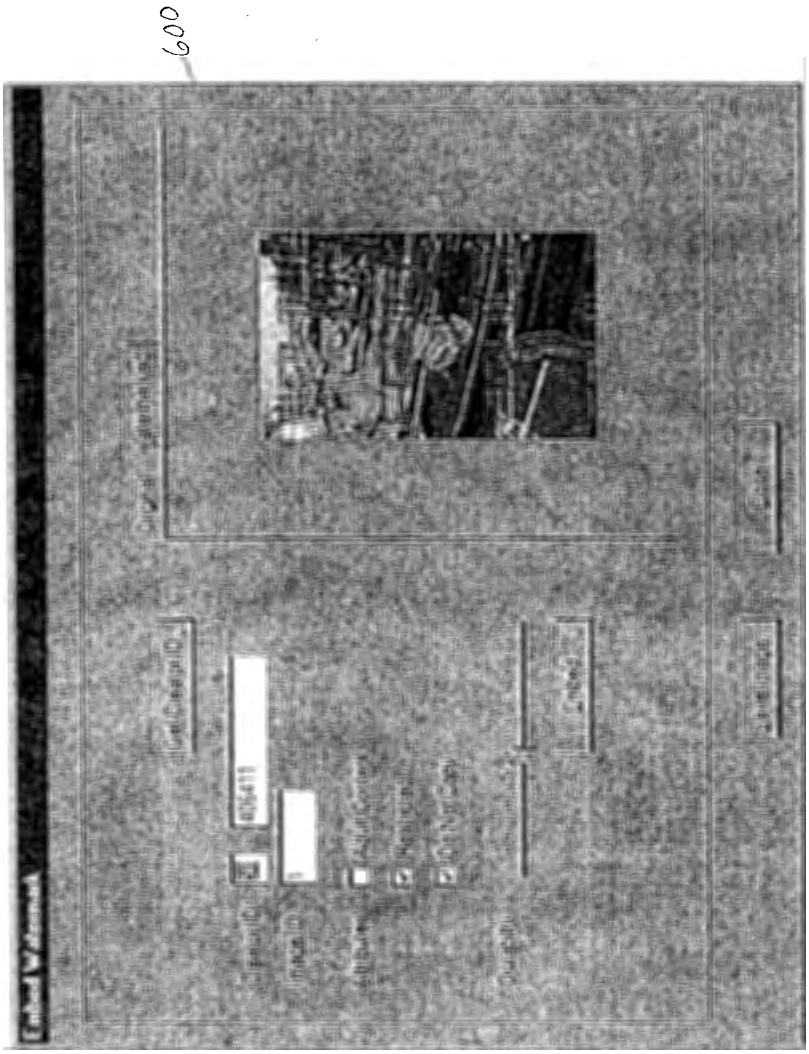


Fig. 6

FIG. 7

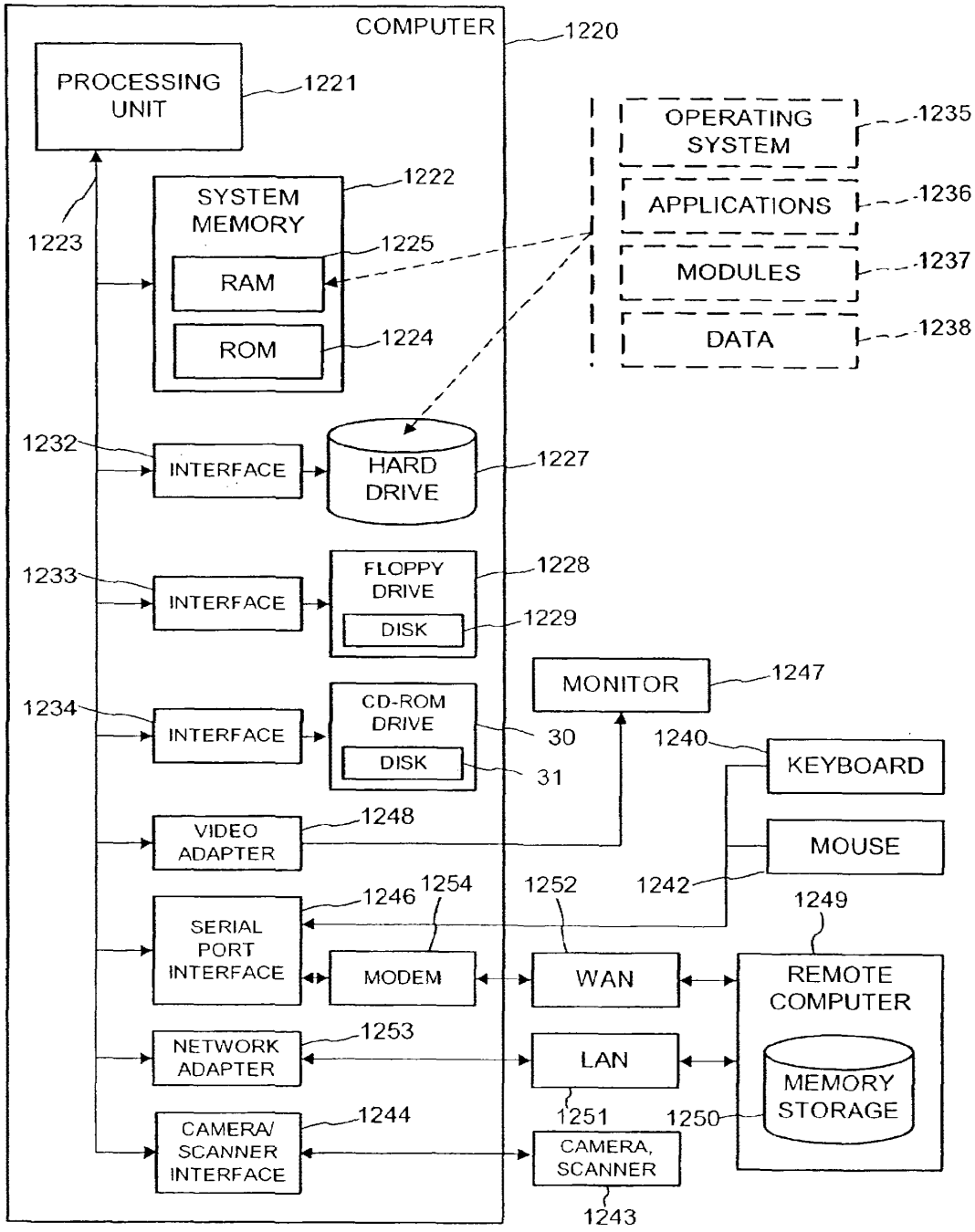
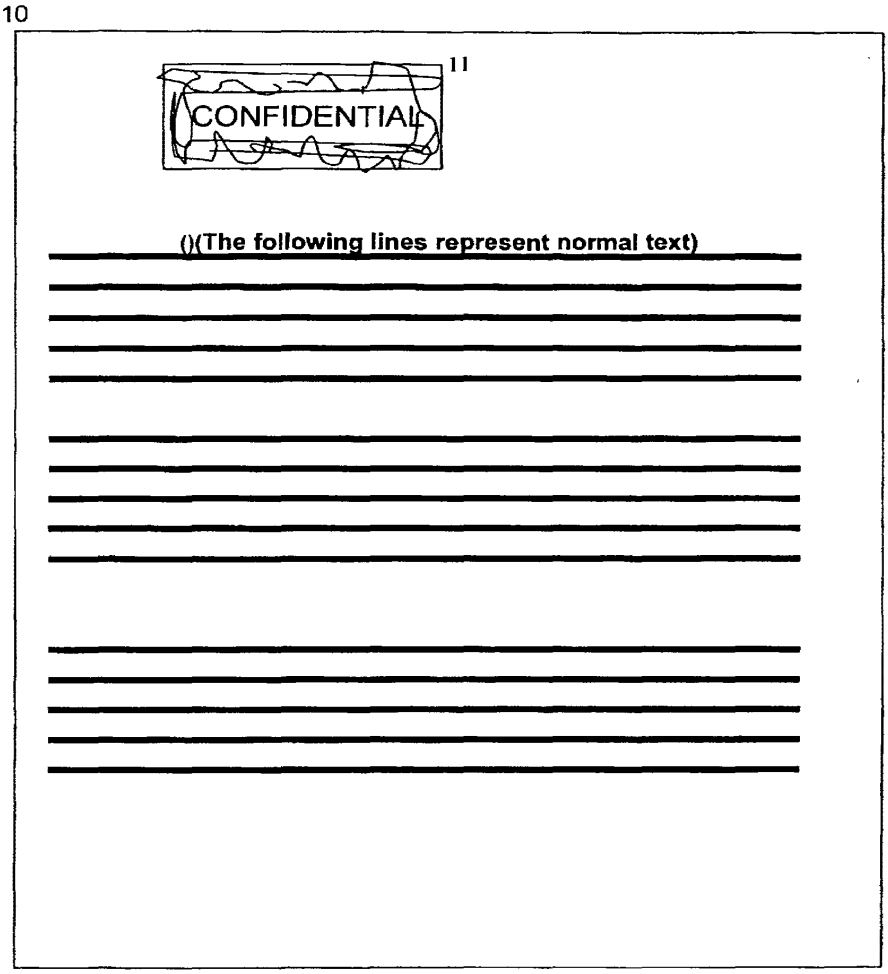

COMPUTER
1220
PROCESSING UNIT
1221
1223
SYSTEM MEMORY
1222
RAM
1225
ROM
1224
OPERATING SYSTEM
1235
APPLICATIONS
1236
MODULES
1237
DATA
1238
1232
INTERFACE
HARD DRIVE
1227
1233
INTERFACE
FLOPPY DRIVE
1228
DISK
1229
1234
INTERFACE
CD-ROM DRIVE
30
DISK
31
MONITOR
1247
1240
KEYBOARD
MOUSE
1242
VIDEO ADAPTER
1248
SERIAL PORT INTERFACE
1246
1254
MODEM
1252
WAN
1249
REMOTE COMPUTER
MEMORY STORAGE
1250
NETWORK ADAPTER
1253
LAN
1251
CAMERA/ SCANNER INTERFACE
1244
CAMERA, SCANNER
1243

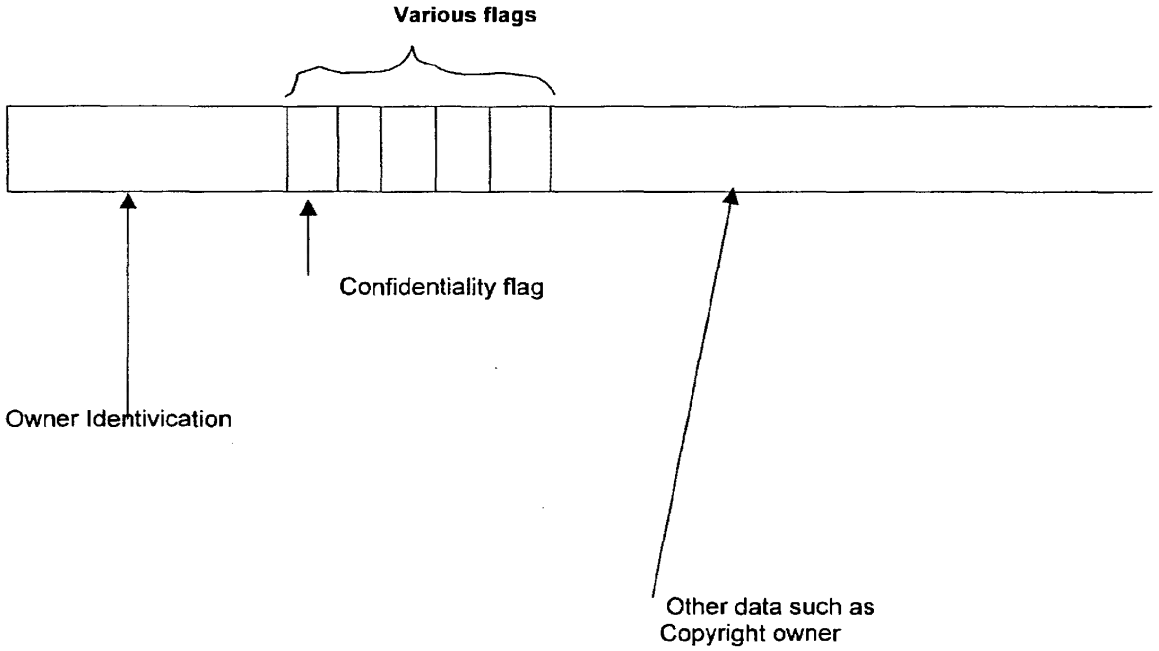
Figure 1 — 8



EWG-076 Figure 1 2-7-00.doc
02/19/00 0

Figure 2 of 9

Fields in a typical watermark payload



EWG-076 Figure 2 2-7-00.doc
02/19/00 11

Figure 3

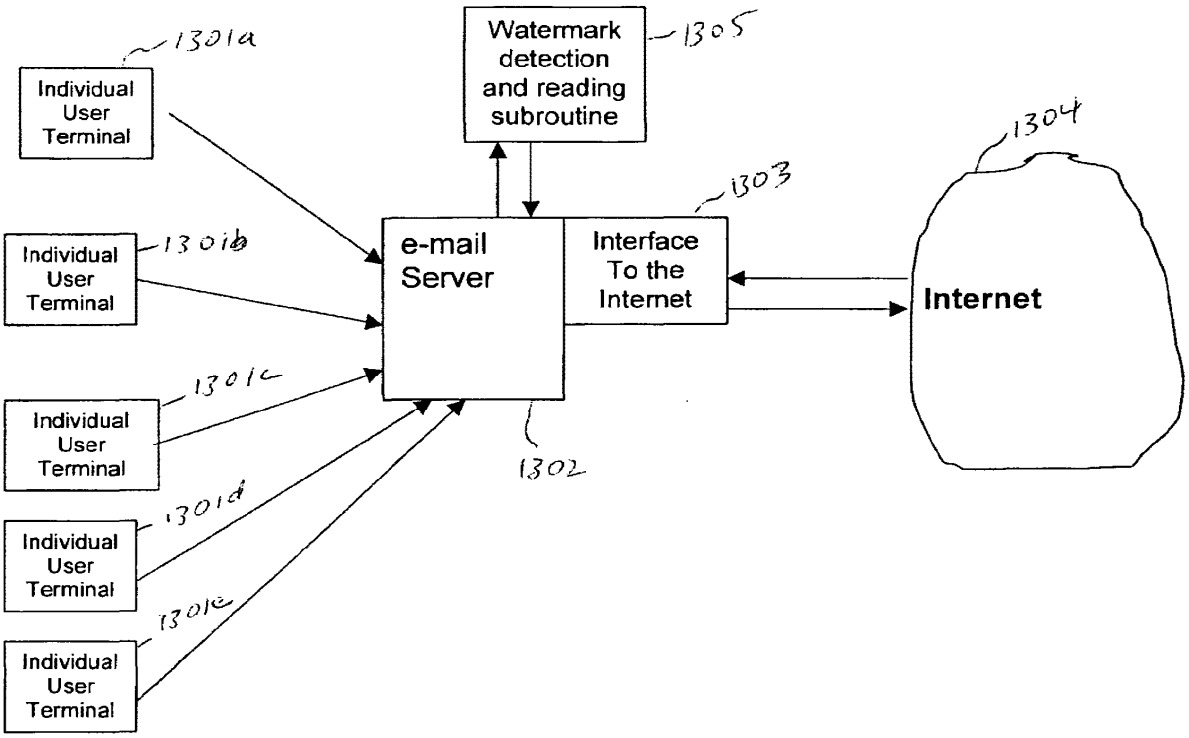

1301a
Individual User Terminal
1301b
Individual User Terminal
1301c
Individual User Terminal
1301d
Individual User Terminal
1301e
Individual User Terminal
Watermark detection and reading subroutine
1305
e-mail Server
Interface To the Internet
1303
1304
Internet
1302

Figure 4:

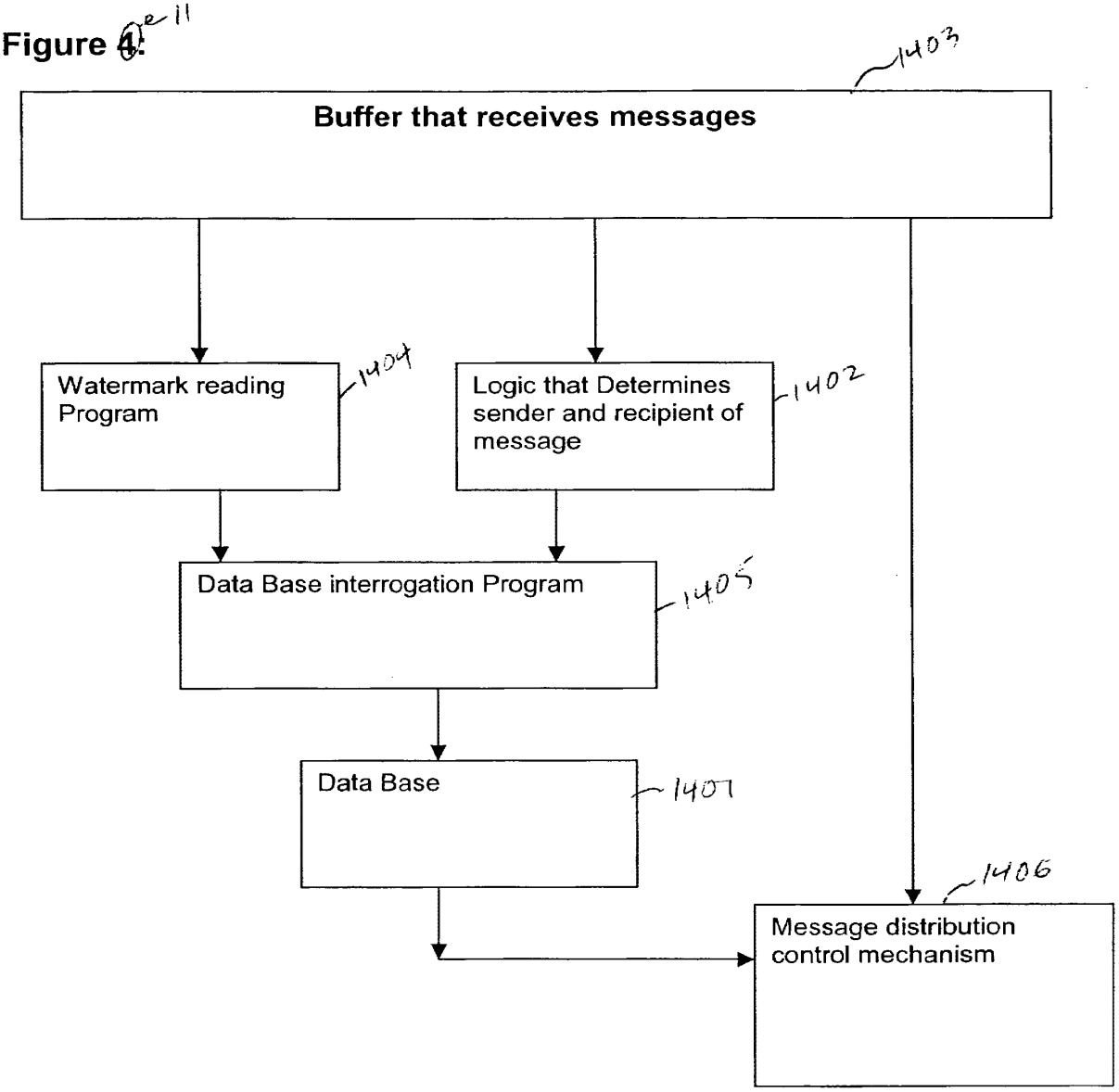
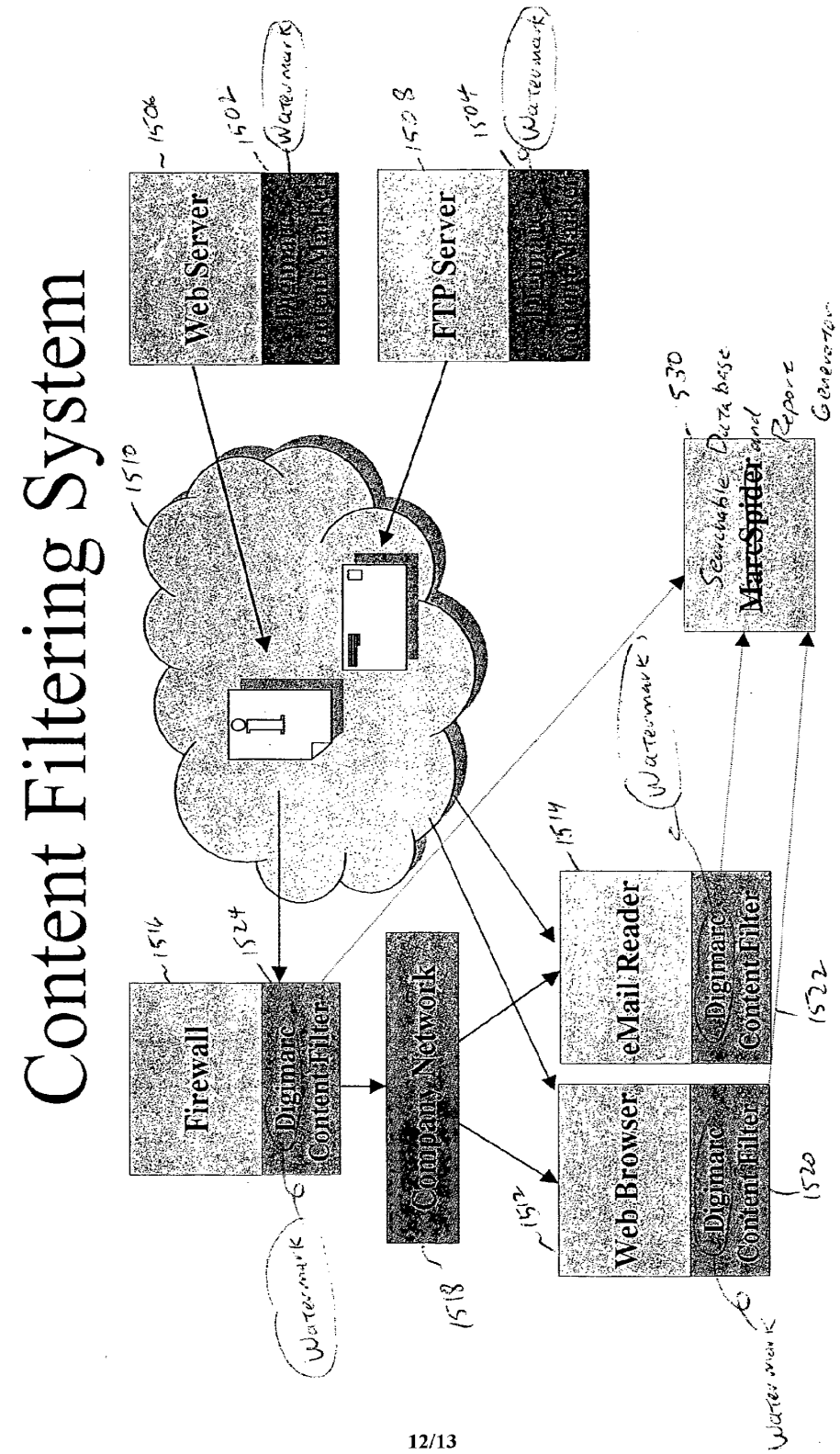

1403
Buffer that receives messages
Watermark reading Program
1404
Logic that Determines sender and recipient of message
1402
Data Base interrogation Program
1405
Data Base
1407
1406
Message distribution control mechanism

Fig. 12


Content Filtering System
Web Server
FTP Server
Firewall
Digimarc Content Filter
Company Network
eMail Reader
Digimarc Content Filter
Web Browser
Digimarc Content Filter
MarcSpider
Searchable Data base and Report Generator
Watermark
1502
1504
1506
1508
1510
1512
1514
1516
1518
1520
1522
1524
1530

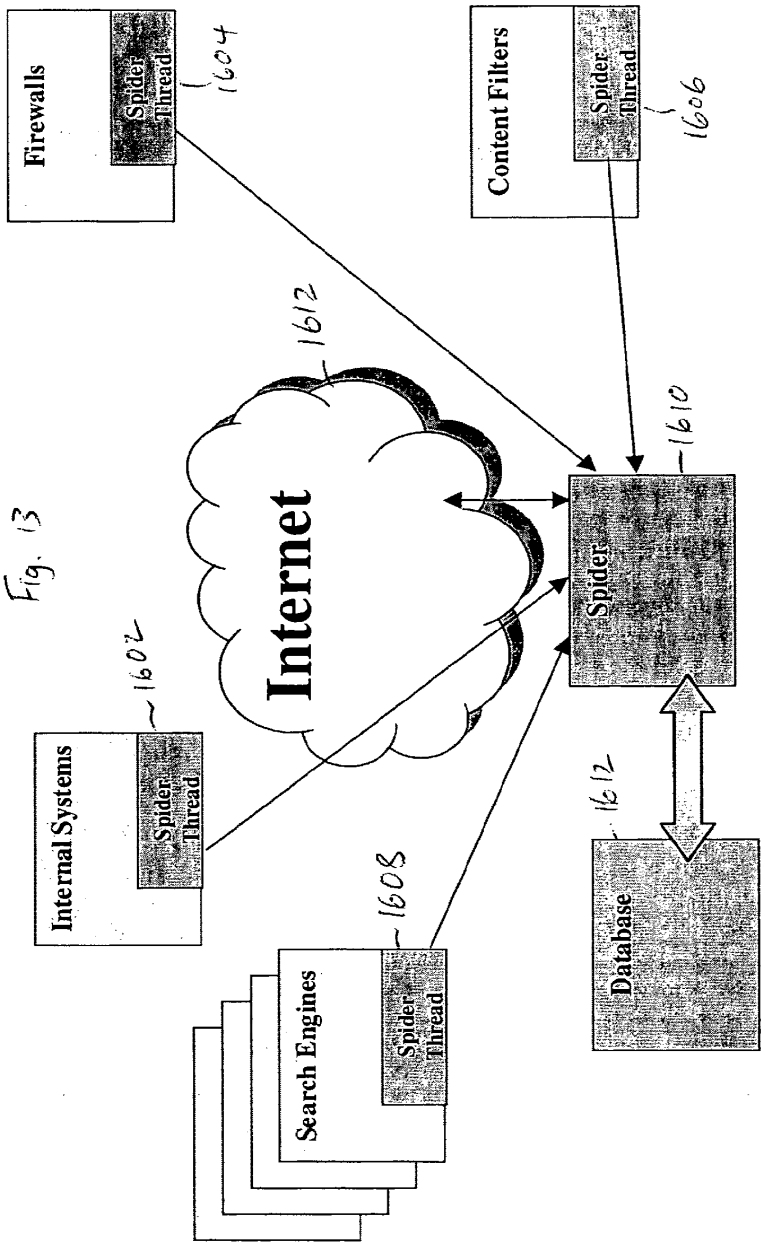

Firewalls
Spider Thread
1604
Content Filters
Spider Thread
1606
1612
Internet
1610
Spider
Internal Systems
Spider Thread
1602
1608
Search Engines
Spider Thread
1612
Database

Fig. 13

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/04812

A. CLASSIFICATION OF SUBJECT MATTER
IPC(7) : G06F 13/00, 15/16; H04L 9/00
US CL : 709/217; 380/28; 345/335
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 709/217; 380/28, 54, 279; 345/335, 339; 283/72, 113; 713/161, 176

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,956,716 A (KENNER et al) 21 SEPTEMBER 1999, column 20, lines 22-column 27, lines 61, column 32, lines 38-column 33, lines 12, column 23, lines 25-49, column 32, lines 64-column 33, lines 12, column 23, lines 66-column 24, lines 14, column 27, lines 64, lines 64-column 28, lines 17, column 29, lines 27-35	1-20
Y	US 5,841,978 A (RHOADS) 24 NOVEMBER 1998, column 2, lines 9-26, column 58, lines 34-column 59, lines 47	1-20

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
01 June 2001 (01.06.2001)

Date of mailing of the international search report
28 JUN2001

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. (703) 305-3230

Authorized officer
Tadesse Hailu

Telephone No. (703) 306-2799

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

tional application No.
PCT/US01/04812

Box I Observations where certain claims were found unsearchable (Continuation of Item 1 of first sheet)

This international report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claim Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claim Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claim Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of Item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:
GROUP 1: CLAIMS 1-20; GROUP II: CLAIMS 21-25; GROUP III: CLAIMS 26-32

The inventions listed as Groups I, II and III do not relate to a single inventive concept under PCT Rule 13.1 because, under PCT Rule 13.2, they lack the same or corresponding special technical features for the following reasons: the special technical feature of Group I invention is the browser system claimed therein while the special technical feature of the Group II invention is the electronic massaging system and the feature of Group III invention is content filtering system. Since the special technical feature of the Group I invention is not present in Groups II and III inventions being claimed and the special technical feature of the Group II invention is not present in Groups I and III inventions being claimed and the special technical feature of the Group III invention is not present in Group I and II inventions being claimed, unity of invention is lacking.

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.: 1-20

Remark on Protest
☐ The additional search fees were accompanied by the applicant's protest.
☐ No protest accompanied the payment of additional search fees.

Form PCT/ISA/210 (continuation of first sheet(1)) (July 1998)

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 October 2001 (04.10.2001)

PCT

(10) International Publication Number
WO 01/73586 A2

(51) International Patent Classification[7]: G06F 17/00

(21) International Application Number: PCT/US01/10151

(22) International Filing Date: 29 March 2001 (29.03.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/537,530 29 March 2000 (29.03.2000) US

(71) Applicant: **DIGITAL:CONVERGENCE CORPORATION** [US/US]; Suite 600, 9101 N. Central Expressway, Dallas, TX 75231 (US).

(72) Inventor: **PHILYAW, Jeffrey, Jovan**; 5968 West Northwest Highway, No. 1813, Dallas, TX 75225 (US).

(74) Agents: **HOWISON, Gregory, M.** et al.; Howison, Chauza, Handley & Arnott, L.L.P., P.O. Box 741715, Dallas, TX 75374-1715 (US).

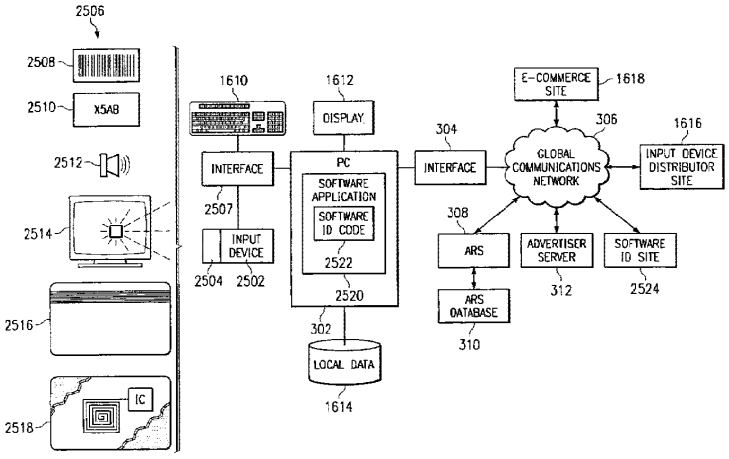
(81) Designated States *(national)*: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD AND APPARATUS FOR ACCESSING A REMOTE LOCATION BY SENSING A MACHINE-RESOLVABLE CODE



(57) Abstract: A method for controlling a computer wherein one or more remote locations disposed on a network are accessed in response to sensing a machine-resolvable code. A computer disposed on a network is operably connected to an input device for sensing a machine-resolvable code. A software application which includes a software identification code runs on the computer. In response to sensing a machine-resolvable code with the input device, the computer accesses at least one remote location corresponding to the software identification code. The one or more remote locations accessed may then return remote information to the computer for presentation.

WO 01/73586 A2

METHOD AND APPARATUS FOR ACCESSING A REMOTE LOCATION BY SENSING A MACHINE-RESOLVABLE CODE

TECHNICAL FIELD OF THE INVENTION

This invention is related to a method of computer control. In one aspect, it relates to a system for automatically directing a web browser application on a computer at a first location to retrieve and display information from a remote location in response to sensing a machine-resolvable code present at the first location.

CROSS REFERENCE TO RELATED APPLICATION

This application is a Continuation-In-Part of pending U.S. Patent Application No. 09/378,221, filed August 19, 1999 and entitled "METHOD AND APPARATUS FOR ACCESSING A REMOTE LOCATION BY SCANNING AN OPTICAL CODE" (Atty Dkt No. PHLY-24,669), which is a Continuation-In-Part of pending U.S. Patent Application Serial No. 09/151,530 and entitled, "METHOD FOR CONTROLLING COMPUTERS THROUGH A RADIO/TELEVISION COMMUNICATION HUB" (Atty Dkt No. PHLY-24,398) filed on September 11, 1998.

BACKGROUND OF THE INVENTION

With the growing numbers of computer users connecting to the "Internet," many companies are seeking the substantial commercial opportunities presented by such a large user base. For example, one technology which exists allows a television ("TV") signal to trigger a computer response in which the consumer will be guided to a personalized web page. The source of the triggering signal may be a TV, video tape recorder, or radio. For example, if a viewer is watching a TV program in which an advertiser offers viewer voting, the advertiser may transmit a unique signal within the television signal which controls a program known as a "browser" on the viewer's computer to automatically display the advertiser's web page. The viewer then simply makes a selection which is then transmitted back to the advertiser.

In order to provide the viewer with the capability of responding to a wide variety of companies using this technology, a database of company information and Uniform Resource Locator ("URL") codes is necessarily maintained in the viewer's computer, requiring continuous updates. URLs are short strings of data that identify resources on the Internet: documents, images, downloadable files, services, electronic mailboxes, and other resources. URLs make resources available under a variety of naming schemes and access methods such as HTTP, FTP, and Internet mail, addressable in the same simple way. URLs reduce the tedium of "login to this server, then issue this magic command ..." down to a single click. The Internet uses URLs to specify the location of files on other servers. A URL includes the type of resource being accessed (e.g., Web, gopher, FTP), the address of the server, and the location of the file. The URL can point to any file on any networked computer. Current technology requires the viewer to perform periodic updates to obtain the most current URL database. This aspect of the current technology is cumbersome since the update process requires downloading information to the viewer's computer. Moreover, the likelihood for error in performing the update, and the necessity of redoing the update in the event of a later computer crash, further complicates the process. Additionally, current technologies are limited in the number of companies which may be stored in the database. This is a significant limitation since world-wide access presented by the Internet and the increasing

number of companies connecting to perform on-line E-commerce necessitates a large database.

SUMMARY OF THE INVENTION

The present invention disclosed and claimed herein comprises, in one aspect thereof, a method for controlling a computer wherein one or more remote locations disposed on a network are accessed in response to sensing a machine-resolvable code. A computer disposed on a network is operably connected to an input device for sensing a machine-resolvable code. A software application which includes a software identification code runs on the computer. In response to sensing a machine-resolvable code with the input device, the computer accesses at least one remote location corresponding to the software identification code. The one or more remote locations accessed may then return remote information to the computer for presentation.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention and the advantages thereof, reference is now made to the following description taken in conjunction with the accompanying Drawings in which:

FIGURE 1 illustrates a block diagram of the preferred embodiment;

FIGURE 2 illustrates the computer components employed in this embodiment;

FIGURE 3 illustrates system interactions over a global network;

FIGURES 4a-4e illustrate the various message packets transmitted between the source PC and network servers used in the preferred embodiment;

FIGURE 5 is a flowchart depicting operation of the system according to the preferred embodiment;

FIGURE 6 illustrates a flowchart of actions taken by the Advertiser Reference Server ("ARS") server;

FIGURE 7 illustrates a flowchart of the interactive process between the source computer and ARS;

FIGURE 8 illustrates a web browser page receiving the modified URL/advertiser product data according to the preferred embodiment;

FIGURE 9 illustrates a simplified block diagram of the disclosed embodiment;

FIGURE 10 illustrates a more detailed, simplified block diagram of the embodiment of FIGURE 9;

FIGURE 11 illustrates a diagrammatic view of a method for performing the routing operation;

FIGURE 12 illustrates a block diagram of an alternate embodiment utilizing an optical region in the video image for generating the routing information;

FIGURE 13 illustrates a block diagram illustrating the generation of a profile with the disclosed embodiment;

FIGURE 14 illustrates a flowchart for generating the profile and storing at the ARS;

FIGURE 15 illustrates a flowchart for processing the profile information when information is routed to a user;

FIGURE 16 illustrates a general block diagram of a disclosed embodiment;

FIGURE 17 illustrates the conversion circuit of the wedge interface;

FIGURE 18 illustrates a sample message packet transmitted from the user PC to the ARS;

FIGURE 19 illustrates a more detailed block diagram of the routing of the message packets between the various nodes;

FIGURE 20 illustrates a block diagram of a browser window, according to a disclosed embodiment;

FIGURE 21 illustrates a diagrammatic view of information contained in the ARS database;

FIGURE 22 illustrates a flowchart of the process of receiving information from the user's perspective;

FIGURE 23 illustrates a flowchart according to the ARS;

FIGURE 24 illustrates a flowchart of the process performed at the E-commerce node;

FIGURE 25 illustrates a general block diagram of another disclosed embodiment;

FIGURE 26 illustrates a sample message packet transmitted from the first PC to the ARS;

FIGURE 27 illustrates another sample message packet transmitted from the first PC to the ARS;

FIGURE 28 illustrates a more detailed block diagram of the routing of the message packets between the various network locations;

FIGURE 29 illustrates a block diagram of a browser window, according to another disclosed embodiment;

FIGURE 30 illustrates a diagrammatic view of information contained in the ARS database according to another disclosed embodiment;

FIGURE 31 illustrates a flowchart of the process of receiving information from the user's perspective;

FIGURE 32 illustrates a flowchart of the process according to the ARS; and

FIGURE 33 illustrates a flowchart of the process performed at the software-specific network location.

DETAILED DESCRIPTION OF THE INVENTION

Referring now to FIGURE 1, there is illustrated a block diagram of a system for controlling a personal computer ("PC") 112 via an audio tone transmitted over a wireless system utilizing a TV. In the embodiment illustrated in FIGURE 1, there is provided a transmission station 101 and a receive station 117 that are connected via a communication link 108. The transmission station 101 is comprised of a television program source 104, which is operable to generate a program in the form of a broadcast signal comprised of video and audio. This is transmitted via conventional techniques along channels in the appropriate frequencies. The program source is input to a mixing device 106, which mixing device is operable to mix in an audio signal. This audio signal is derived from an audio source 100 which comprises a coded audio signal which is then modulated onto a carrier which is combined with the television program source 104. This signal combining can be done at the audio level, or it can even be done at the RF level in the form of a different carrier. However, the preferred method is to merely sum the audio signal from the modulator 102 into the audio channel of the program that is generated by the television program source 104. The output thereof is provided from the mixing device 106 in the form of broadcast signal to an antenna 107, which transmits the information over the communication link 108 to an antenna 109 on the receive side.

On the receive side of the system, a conventional receiver 110, such as a television is provided. This television provides a speaker output which provides the user with an audible signal. This is typically associated with the program. However, the receiver 110 in the disclosed embodiment, also provides an audio output jack, this being the type RCA jack. This jack is utilized to provide an audio output signal on a line 113 which is represented by an audio signal 111. This line 113 provides all of the audio that is received over the communication link 108 to the PC 112 in the audio input port on the PC 112. However, it should be understood that, although a direct connection is illustrated from the receiver 110 to the PC 112, there actually could be a microphone pickup at the PC 112 which could pick the audio signal up. In the disclosed embodiment, the audio signal generated by the

advertiser data input device 100 is audible to the human ear and, therefore, can be heard by the user. Therefore, no special filters are needed to provide this audio to the PC 112.

The PC 112 is operable to run programs thereon which typically are stored in a program file area 116. These programs can be any type of programs such as word processing programs, application programs, etc. In the disclosed embodiment, the program that is utilized in the system is what is referred to as a "browser." The PC 112 runs a browser program to facilitate the access of information on the network, for example, a global communication network known as the "Internet" or the World-Wide-Web ("Web"). The browser is a hypertext-linked application used for accessing information. Hypertext is a term used to describe a particular organization of information within a data processing system, and its presentation to a user. It exploits the computer's ability to link together information from a wide variety of sources to provide the user with the ability to explore a particular topic. The traditional style of presentation used in books employs an organization of the information which is imposed upon it by limitations of the medium, namely fixed sized, sequential paper pages. Hypertext systems, however, use a large number of units of text or other types of data such as image information, graphical information, video information, or sound information, which can vary in size. A collection of such units of information is termed a hypertext document, or where the hypertext documents employ information other than text, hypermedia documents. Multimedia communications may use the Hypertext Transfer Protocol ("HTTP"), and files or formatted data may use the Hypertext Markup Language ("HTML"). This formatting language provides for a mingling of text, graphics, sound, video, and hypertext links by "tagging" a text document using HTML. Data encoded using HTML is often referred to as an "HTML document," an "HTML page," or a "home page." These documents and other Internet resources may be accessed across the network by means of a network addressing scheme which uses a locator referred to as a Uniform Resource Locator ("URL"), for example, "http://www.digital.com."

The Internet is one of the most utilized networks for interconnecting distributed computer systems and allows users of these computer systems to exchange data all over the world. Connected to the Internet are many private networks, for example, corporate or

commercial networks. Standard protocols, such as the Transport Control Protocol ("TCP") and the Internet Protocol ("IP") provide a convenient method for communicating across these diverse networks. These protocols dictate how data are formatted and communicated. As a characteristic of the Internet, the protocols are layered in an IP stack. At higher levels of the IP stack, such as the application layer (where HTTP is employed), the user information is more readily visible, while at lower levels, such as the network level (where TCP/IP are used), the data can merely be observed as packets or a stream of rapidly moving digital signals. Superimposed on the Internet is a standard protocol interface for accessing Web resources, such as servers, files, Web pages, mail messages, and the like. One way that Web resources can be accessed is by browsers made by *Netscape*® and *Microsoft Internet Explorer*®.

Referring again now to FIGURE 1, the user can load this program with the appropriate keystrokes such that a browser window will be displayed on a display 118. In one embodiment, the user can run the browser program on the PC 112 such that the browser window is displayed on the display 118. While watching a preferred program, the user can also view display 118. When an audio signal is received by the receiver 110 and the encoded information is contained therein that was input thereto by the advertiser, the PC 112 will then perform a number of operations. The first operation, according to the disclosed embodiment, is to extract the audio information within the received audio signal in the form of digital data, and then transmit this digital data to a defined location on the global communication network via a modem connection 114. This connection will be described hereinbelow. This information will be relayed to a proprietary location and the instructions sent back to the PC 112 as to the location of the advertiser associated with the code, and the PC 112 will then effect a communication link to that location such that the user can view on the display 118 information that the advertiser, by the fact of putting the tone onto the broadcast channel, desires the viewer to view. This information can be in the form of interactive programs, data files, etc. In one example, when an advertisement appears on the television, the tone can be generated and then additional data displayed on the display 118. Additionally, a streaming video program could be played on the PC received over the network, which streaming video program is actually longer than the

advertising segment on the broadcast. Another example would be a sports game that would broadcast the tone in order to allow a user access to information that is not available over the broadcast network, such as additional statistics associated with the sports program, etc.

By utilizing the system described herein with respect to the disclosed embodiment of FIGURE 1, an advertiser is allowed the ability to control a user's PC 112 through the use of tones embedded within a program audio signal. As will described hereinbelow, the disclosed embodiment utilizes particular routing information stored in the PC 112 which allows the encoded information in the received audio signal to route this information to a desired location on the network, and then allow other routing information to be returned to the PC 112 for control thereof to route the PC 112 to the appropriate location associated with that code.

Referring now to FIGURE 2, there is illustrated a computer 204, similar to computer 112, connected to display information on display 118. The computer 204 comprises an internal audio or "sound" card 206 for receiving the transmitted audio signal through receive antenna 109 and receiver 110. The sound card 206 typically contains analog-to-digital circuitry for converting the analog audio signal into a digital signal. The digital signal may then be more easily manipulated by software programs. The receiver 110 separates the audio signal from the video signal. A special trigger signal located within the transmitted advertiser audio signal triggers proprietary software running on the computer 204 which launches a communication application, in this particular embodiment, the web browser application located on the PC 204. Coded advertiser information contained within the audio signal is then extracted and appended with the address of a proprietary server located on the communication network.

The remote server address is in the form of a URL. This appended data, in addition to other control codes, is inserted directly into the web browser application for automatic routing to the communication network. The web browser running on PC 204, and communicating to the network with an internal modem 208, in this embodiment, transmits the advertiser information to the remote server. The remote server cross-references the

advertiser product information to the address of the advertiser server located on the network. The address of the advertiser server is routed back through the PC 204 web browser to the advertiser server. The advertiser product information is returned to PC 204 to be presented to the viewer on display 118. In this particular embodiment, the particular advertiser product information displayed is contained within the advertiser's web page 212. As mentioned above, the audio signal is audible to the human ear. Therefore the audio signal, as emitted from the TV speakers, may be input to the sound card 206 via a microphone. Furthermore, the audio signal need not be a real-time broadcast, but may be on video tapes, CDs, DVD, or other media which may be displayed at a later date. With the imminent implementation of high definition digital television, the audio signal output from the TV may also be digital. Therefore, direct input into a sound card for A/D purposes may not be necessary, but alternative interfacing techniques to accommodate digital-to-digital signal formats would apply.

Referring now to FIGURE 3, there is illustrated a source PC 302, similar to PCs 204 and 112, connected to a global communication network ("GCN") 306 through an interface 304. In this embodiment, the audio signal 111 is received by PC 302 through its sound card 206. The audio signal 111 comprises a trigger signal which triggers proprietary software into launching a web browser application residing on the PC 302. The audio signal 111 also comprises advertiser product information which is extracted and appended with URL information of an Advertiser Reference Server ("ARS") 308. The ARS 308 is a system disposed on the GCN 306 that is defined as the location to which data in the audio signal 111 is to be routed. As such, data in the audio signal 111 will always be routed to the ARS 308, since a URL is unique on the GCN 306. Connected to the ARS 308 is a database 310 of product codes and associated manufacturer URLs. The database 310 undergoes a continual update process which is transparent to the user. As companies sign-on, i.e., subscribe, to this technology, manufacturer and product information is added to the database 310 without interrupting operation of the source PC 302 with frequent updates. When the advertiser server address URL is obtained from the ARS database 310, it and the request for the particular advertiser product information are automatically routed back through the web browser on PC 302, over to the respective advertiser server for retrieval of the advertiser

product information to the PC 302. Additionally, although the disclosed invention discusses a global communication network, the system is also applicable to LANs, WANs, and peer-to-peer network configurations. It should be noted that the disclosed architecture is not limited to a single source PC 302, but may comprise a plurality of source PCs, e.g., PC 300 and PC 303. Moreover, a plurality of ARS 308 systems and advertiser servers 312 may be implemented, e.g., ARS 314, and advertiser server A 316, respectively.

The information transactions, in general, which occur between the networked systems of this embodiment, over the communication network, are the following. The web browser running on source PC 302 transmits a message packet to the ARS 308 over Path "A." The ARS 308 decodes the message packet and performs a cross-reference function with product information extracted from the received message packet to obtain the address of an advertiser server 312. A new message packet is assembled comprising the advertiser server 312 address, and sent back to the source PC 302 over Path "B." A "handoff" operation is performed whereby the source PC 302 browser simply reroutes the information on to the advertiser server 312 over Path "C," with the appropriate source and destination address appended. The advertiser server 312 receives and decodes the message packet. The request-for-advertiser-product-information is extracted and the advertiser 312 retrieves the requested information from its database for transmission back to the source PC 302 over Path "D." The source PC 302 then processes the information, i.e., for display to the viewer. The optional Path "E" is discussed hereinbelow. It should be noted that the disclosed methods are not limited to only browser communication applications, but may accommodate, with sufficient modifications by one skilled in the art, other communication applications used to transmit information over the Internet or communication network.

Referring now to FIGURE 4a, the message packet 400 sent from the source PC 302 to ARS 308 via Path "A" comprises several fields. One field comprises the URL of the ARS 308 which indicates where the message packet is to be sent. Another field comprises the advertiser product code or other information derived from the audio signal 111, and any additional overhead information required for a given transaction. The product code provides a link to the address of the advertiser server 312, located in the database 310. Yet

another field comprises the network address of the source PC 302. In general, network transmissions are effected in packets of information, each packet providing a destination address, a source address, and data. These packets vary depending upon the network transmission protocol utilized for communication. Although the protocols utilized in the disclosed embodiment are of a conventional protocol suite commonly known as TCP/IP, it should be understood that any protocols providing the similar basic functions can be used, with the primary requirement that a browser can forward the routing information to the desired URL in response to keystrokes being input to a PC. Within the context of this disclosure, "message packet" shall refer to and comprise the destination URL, product information, and source address, even though more than a single packet must be transmitted to effect such a transmission.

Upon receipt of the message packet 400 from source PC 302, ARS 308 processes the information in accordance with instructions embedded in the overhead information. The ARS 308 specifically will extract the product code information from the received packet 400 and, once extracted, will then decode this product code information. Once decoded, this information is then compared with data contained within the ARS advertiser database 310 to determine if there is a "hit." If there is no "hit" indicating a match, then information is returned to the browser indicating such. If there is a "hit," a packet 402 is assembled which comprises the address of the source PC 302, and information instructing the source PC 302 as to how to access, directly in a "handoff" operation, another location on the network, that of an advertiser server 312. This type of construction is relatively conventional with browsers such as *Netscape*® and *Microsoft Internet Explorer*® and, rather than displaying information from the ARS 308, the source PC 302 can then access the advertiser server 312. The ARS 308 transmits the packet 402 back to source PC 302 over Path "B." Referring now to FIGURE 4b, the message packet 402 comprises the address of the source PC 302, the URL of the advertiser server 312 embedded within instructional code, and the URL of the ARS 308.

Upon receipt of the message packet 402 by the source PC 302, the message packet 402 is disassembled to obtain pertinent routing information for assembly of a new message

packet 404. The web browser running on source PC 302 is now directed to obtain, over Path "C," the product information relevant to the particular advertiser server 312 location information embedded in message packet 404. Referring now to FIGURE 4c, the message packet 404 for this transaction comprises the URL of the advertiser server 312, the request-for-product-information data, and the address of the source PC 302.

Upon receipt of the message packet 404 from source PC 302, advertiser server 312 disassembles the message packet 404 to obtain the request-for-product-information data. The advertiser server 312 then retrieves the particular product information from its database, and transmits it over Path "D" back to the source PC 302. Referring now to FIGURE 4d, the message packet 406 for this particular transaction comprises the address of the source PC 302, the requested information, and the URL of the advertiser server 312.

Optionally, the ARS 308 may make a direct request for product information over Path "E" to advertiser server 312. In this mode, the ARS 308 sends information to the advertiser server 312 instructing it to contact the source PC 302. This, however, is unconventional and requires more complex software control. The message packet 408 for this transaction is illustrated in FIGURE 4e, which comprises the URL of the advertiser server 312, the request-for-product-information data, and the address of the source PC 302. Since product information is not being returned to the ARS 308, but directly to the source PC 302, the message packet 408 requires the return address to be that of the source PC 302. The product information is then passed directly to PC 302 over Path "D."

Referring now to FIGURE 5, the method for detecting and obtaining product information is as follows. In decision block 500, a proprietary application running resident on a source computer PC 302 (similar to PC 204) monitors the audio input for a special trigger signal. Upon detection of the trigger signal, data following the trigger signal is decoded for further processing, in function block 502. In function block 504, the data is buffered for further manipulation. In decision block 506, a determination is made as to whether the data can be properly authenticated. If not, program flow continues through the "N" signal to function block 520 where the data is discarded. In function block 522, the

program then signals for a retransmission of the data. The system then waits for the next trigger signal, in decision block 500. If properly authenticated in decision block 506, program flow continues through the "Y" signal path where the data is then used to launch the web browser application, as indicated in function block 508. In function block 510, the web browser receives the URL data, which is then automatically routed through the computer modem 208 to the network interface 304 and ultimately to the network 306. In function block 514, the ARS 308 responds by returning the URL of advertiser server 312 to the PC 302. In function block 516, the web browser running on the source PC 302, receives the advertiser URL information from the ARS 308, and transmits the URL for the product file to the advertiser server 312. In block 518, the advertiser server 312 responds by sending the product information to the source PC 302 for processing.

The user may obtain the benefits of this architecture by simply downloading the proprietary software over the network. Other methods for obtaining the software are well-known; for example, by CD, diskette, or pre-loaded hard drives.

Referring now to FIGURE 6, there is illustrated a flowchart of the process the ARS 308 may undergo when receiving the message packet 400 from the source PC 302. In decision block 600, the ARS 308 checks for the receipt of the message packet 400. If a message packet 400 is not received, program flow moves along the "N" path to continue waiting for the message. If the message packet 400 is received, program flow continues along path "Y" for message processing. Upon receipt of the message packet 400, in function block 602, the ARS 308 decodes the message packet 400. The product code is then extracted independently in function block 604 in preparation for matching the product code with the appropriate advertiser server address located in the database 310. In function block 606, the product code is then used with a lookup table to retrieve the advertiser server 312 URL of the respective product information contained in the audio signal data. In function block 608, the ARS 308 then assembles message packet 402 for transmission back to the source PC 302. Function block 610 indicates the process of sending the message packet 402 back to the source PC 302 over Path "B."

Referring now to FIGURE 7, there is illustrated a flowchart of the interactive processes between the source PC 302 and the advertiser server 312. In function block 700, the source PC 302 receives the message packet 402 back from the ARS 308 and begins to decode the packet 402. In function block 702, the URL of the advertiser product information is extracted from the message packet 402 and saved for insertion into the message packet 404 to the advertiser server 312. The message packet 404 is then assembled and sent by the source PC 302 over Path "C" to the advertiser server 312, in function block 704. While the source PC 302 waits, in function block 706, the advertiser server 312 receives the message packet 404 from the source PC 302, in function block 708, and disassembles it. The product information location is then extracted from the message packet 404 in function block 710. The particular product information is retrieved from the advertiser server 312 database for transmission back to the source PC 302. In function block 712, the product information is assembled into message packet 406 and then transmitted back to the source PC 302 over Path "D." Returning to the source PC 302 in function block 714, the advertiser product information contained in the message packet 406 received from the advertiser server 312, is then extracted and processed in function block 716.

Referring now to FIGURE 8, after receipt of a trigger signal, a web browser application on a source PC 302 is automatically launched and computer display 800 presents a browser page 802. Proprietary software running on the source PC 302 processes the audio signal data after being digitized through the sound card 206. The software appropriately prepares the data for insertion directly into the web browser by extracting the product information code and appending keystroke data to this information. First, a URL page 804 is opened in response to a Ctrl-O command added by the proprietary software as the first character string. Opening URL page 804 automatically positions the cursor in a field 806 where additional keystroke data following the Ctrl-O command will be inserted. After URL page 804 is opened, the hypertext protocol preamble http:// is inserted into the field 806. Next, URL information associated with the location of the ARS 308 is inserted into field 806. Following the ARS 308 URL data are the characters /? to allow entry of variables immediately following the /? characters. In this embodiment, the variable following is the

product information code received in the audio signal. The product code information also provides the cross-reference information for obtaining the advertiser URL from the ARS database 310. Next, a carriage return is added to send the URL/product data and close the window 804. After the message packet 400 is transmitted to the ARS 308 from the source PC 302, transactions from the ARS308, to the source PC 302, to the advertiser server 312, and back to the source PC 302, occur quickly and are transparent to the viewer. At this point, the next information the viewer sees is the product information which was received from the advertiser server 312.

Referring now to FIGURE 9, there is illustrated a block diagram of a more simplified embodiment. In this embodiment, a video source 902 is provided which is operable to provide an audio output on an audio cable 901 which provides routing information referred to by reference numeral 904. The routing information 904 is basically information contained within the audio signal. This is an encoded or embedded signal. The important aspect of the routing information 904 is that it is automatically output in realtime as a function of the broadcast of the video program received over the video source 902. Therefore, whenever the program is being broadcast in realtime to the user 908, the routing information 904 will be output whenever the producer of the video desires it to be produced. It should be understood that the box 902 representing the video source could be any type of media that will result in the routing information being output. This could be a cassette player, a DVD player, an audio cassette, a CD ROM or any such media. It is only important that this is a program that the producer develops which the user 908 watches in a continuous or a streaming manner. Embedded within that program, at a desired point selected by the producer, the routing information 904 is output.

The audio information is then routed to a PC 906, which is similar to the PC 112 in FIGURE 1. A user 908 is interfaced with the PC to receive information thereof, the PC 906 having associated therewith a display (not shown). The PC 906 is interfaced with a network 910, similar to the network 306 in FIGURE 3. This network 910 has multiple nodes thereon, one of which is the PC 906, and another of which is represented by a network node 912 which represents remote information. The object of the present embodiment is to

access remote information for display to the user 908 by the act of transmitting from the video program in block 902 the routing information 904. This routing information 904 is utilized to allow the PC 906 which has a network "browser" running thereon to "fetch" the remote information at the node 912 over the network 910 for display to the user 908. This routing information 904 is in the form of an embedded code within the audio signal, as was described hereinabove.

Referring now to FIGURE 10, there is illustrated a more detailed block diagram of the embodiment of FIGURE 9. In this embodiment, the PC 906 is split up into a couple of nodes, a first PC 1002 and a second PC 1004. The PC 1002 resides at the node associated with the user 908, and the PC 1004 resides at another node. The PC 1004 represents the ARS 308 of FIGURE 3. The PC 1004 has a database 1006 associated therewith, which is basically the advertiser database 310. Therefore, there are three nodes on the network 910 necessary to implement the disclosed embodiment, the PC 1002, the PC 1004 and the remote information node 912. The routing information 904 is utilized by the PC 1002 for routing to the PC 1004 to determine the location of the remote information node 912 on the network 910. This is returned to the PC 1002 and a connection made directly with the remote information node 912 and the information retrieved therefrom to the user 908. The routing information 904 basically constitutes primary routing information.

Referring now to FIGURE 11, there is illustrated a diagrammatic view of how the network packet is formed for sending the primary routing information to the PC 1004. In general, the primary routing information occupies a single field which primary routing information is then assembled into a data packet with the secondary routing information for transfer to the network 910. This is described hereinabove in detail.

Referring now to FIGURE 12, there is illustrated an alternate embodiment to that of FIGURE 9. In this embodiment, the video source 902 has associated therewith an optical region 1202, which optical region 1202 has disposed therein an embedded video code. This embedded video code could be relatively complex or as simple as a grid of dark and white regions, each region in the grid able to have a dark color for a logic "1" or a white region for

a logic "0." This will allow a digital value to be disposed within the optical region 1202. A sensor 1204 can then be provided for sensing this video code. In the example above, this would merely require an array of optical detectors, one for each region in the grid to determine whether this is a logic "1" or a logic "0" state. One of the sensed video is then output to the PC 906 for processing thereof to determine the information contained therein, which information contained therein constitutes the primary routing information 904. Thereafter, it is processed as described hereinabove with reference to FIGURE 9.

Referring now to FIGURE 13, there is illustrated a block diagram for an embodiment wherein a user's profile can be forwarded to the original subscriber or manufacturer. The PC 906 has associated therewith a profile database 1302, which profile database 1302 is operable to store a profile of the user 908. This profile is created when the program, after initial installation, requests profile information to be input in order to activate the program. In addition to the profile, there is also a unique ID that is provided to the user 908 in association with the browser program that runs on the PC 906. This is stored in a storage location represented by a block 1304. This ID 1304 is accessible by a remote location as a "cookie" which is information that is stored in the PC 906 in an accessible location, which accessible location is actually accessible by the remote program running on a remote node.

The ARS 308, which basically constitutes the PC 1004 of FIGURE 10, is operable to have associated therewith a profile database 1308, which profile database 1308 is operable to store profiles for all of the users. The profile database 1308 is a combination of the stored in profile database 1302 for all of the PCs 906 that are attachable to the system. This is to be distinguished from information stored in the database 310 of the ARS 308, the advertiser's database, which contains intermediate destination tables. When the routing information in the primary routing information 904 is forwarded to the ARS 308 and extracted from the original data packet, the lookup procedure described hereinabove can then be performed to determine where this information is to be routed. The profile database 1302 is then utilized for each transaction, wherein each transaction in the form of the

routing information received from the primary routing information 904 is compared to the destination tables of database 310 to determine what manufacturer is associated therewith.

The associated ID 1304 that is transmitted along with the routing information in primary routing information 904 is then compared with the profile database 1308 to determine if a profile associated therewith is available. This information is stored in a transaction database 1310 such that, at a later time, for each routing code received in the form of the information in primary routing information 904, there will associated therewith the IDs 1304 of each of the PCs 906. The associated profiles in database 1308, which are stored in association with IDs 1304, can then be assembled and transmitted to a subscriber as referenced by a subscriber node 1312 on the network 910. The ARS 308 can do this in two modes, a realtime mode or a non-realtime mode. In a realtime mode, each time a PC 906 accesses the advertiser database 310, that user's profile information is uploaded to the subscriber node 1312. At the same time, billing information is generated for that subscriber 1312 which is stored in a billing database 1316. Therefore, the ARS 308 has the ability to inform the subscriber 1312 of each transaction, bill for those transactions, and also provide to the subscriber 1312 profile information regarding who is accessing the particular product advertisement having associated therewith the routing information field 904 for a particular routing code as described hereinabove. This information, once assembled, can then be transmitted to the subscriber 1312 and also be reflected in billing information and stored in the billing information database 1316.

Referring now to FIGURE 14, there is illustrated a flowchart depicting the operation for storing the profile for the user. The program is initiated in a block 1402 and then proceeds to a function block 1404, wherein the system will prompt for the profile upon initiation of the system. This initiation is a function that is set to activate whenever the user initially loads the software that he or she is provided. The purpose for this is to create, in addition to the setup information, a user profile. Once the user is prompted for this, then the program will flow to a decision block 1406 to determine whether the user provides basic or detailed information. This is selectable by the user. If selecting basic, the program will flow to a function block 1408 wherein the user will enter basic information such as name

and serial number and possibly an address. However, to provide some incentive to the user to enter more information, the original prompt in function block 1404 would have offers for such things as coupons, discounts, etc., if the user will enter additional information. If the user selects this option, the program flows from the decision block 1406 to a function block 1410. In the function block 1410, the user is prompted to enter specific information such as job, income level, general family history, demographic information and more. There can be any amount of information collected in this particular function block.

Once all of the information is collected, in either the basic mode or the more specific mode, the program will then flow to a function block 1412 where this information is stored locally. The program then flows to a decision block 1414 to then go on-line to the host or the ARS 308. In general, the user is prompted to determine whether he or she wants to send this information to the host at the present time or to send it later. If he or she selects the "later" option, the program will flow to a function block 1415 to prompt the user at a later time to send the information. In the disclosed embodiment, the user will not be able to utilize the software until the profile information is sent to the host. Therefore, the user may have to activate this at a later time in order to connect with the host.

If the user has selected the option to upload the profile information to the host, the program will flow to the function block 1416 to initiate the connect process and then to a decision block 1418 to determine if the connection has been made. If not, the program will flow along a "N" path to a time to decision block 1420 which will timeout to an error block 1422 or back to the input of the connect decision block 1418. The program, once connected, will then flow along a "Y" path from decision block 1418 to a function block 1428 to send the profile information with the ID of the computer or user to the host. The ID is basically, as described hereinabove, a "cookie" in the computer which is accessed by the program when transmitting to the host. The program will then flow to a function block 1430 to activate the program such that it, at later time, can operate without requiring all of the setup information. In general, all of the operation of this flowchart is performed with a "wizard" which steps the user through the setup process. Once complete, the program will flow to a Done block 1432.

Referring now to FIGURE 15, there is illustrated a flowchart depicting the operation of the host when receiving a transaction. The program is initiated at a Start block 1502 and then proceeds to decision block 1504, wherein it is determined whether the system has received a routing request, i.e., the routing information 904 in the form of a tone, etc., embedded in the audio signal, as described hereinabove with respect to FIGURE 9. The program will loop back around to the input of decision block 1504 until the routing request has been received. At this time, the program will flow along the "Y" path to a function block 1506 to receive the primary routing information and the user ID. Essentially, this primary routing information is extracted from the audio tone, in addition to the user ID. The program then flows to a function block 1508 to look up the manufacturer URL that corresponds to the received primary routing information and then return the necessary command information to the originating PC 108 in order to allow that PC 108 to connect to the destination associated with the primary routing information. Thereafter, the program will flow to a function block 1510 to update the transaction database 1310 for the current transaction. In general, the routing information 904 will be stored as a single field with the associated IDs. The profile database 1308, as described hereinabove, has associated therewith detailed profiles of each user on the system that has activated their software in association with their ID. Since the ID was sent in association with the routing information, what is stored in the transaction database 1310 is the routing code, in association with all of the IDs transmitted to the system in association with that particular routing code. Once this transaction database 1310 has been updated, as described hereinabove, the transactions can be transferred back to the subscriber at node 312 with the detailed profile information from the profile database 1308.

The profile information can be transmitted back to the subscriber or manufacturer at the node 312 in realtime or non-realtime. A decision block 1512 is provided for this, which determines if the delivery is realtime. If realtime, the program will flow along a "Y" path to a function block 1514 wherein the information will be immediately forwarded to the manufacturer or subscriber. The program will then flow to a function block 1516 wherein the billing for that particular manufacturer or subscriber will be updated in the billing database 1316. The program will then flow into an End block 1518. If it was non-realtime,

the program moves along the "N" path to a function block 1520 wherein it is set for a later delivery and it is accrued in the transaction database 1310. In any event, the transaction database 1310 will accrue all information associated with a particular routing code.

With a realtime transaction, it is possible for a manufacturer to place an advertisement in a magazine or to place a product on a shelf at a particular time. The manufacturer can thereafter monitor the times when either the advertisements are or the products are purchased. Of course, they must be scanned into a computer which will provide some delay. However, the manufacturer can gain a very current view of how a product is moving. For example, if a cola manufacturer were to provide a promotional advertisement on, for example, television, indicating that a new cola was going to be placed on the shelf and that the first 1000 purchasers, for example, scanning their code into the network would receive some benefit, such as a chance to win a trip to some famous resort in Florida or some other incentive, the manufacturer would have a very good idea as to how well the advertisement was received. Further, the advertiser would know where the receptive markets were. If this advertiser, for example, had placed the television advertisement in ten cities and received overwhelming response from one city, but very poor response from another city, he would then have some inclination to believe that either the one poor-response city was not a good market or that the advertising medium he had chosen was very poor. Since the advertiser can obtain a relatively instant response and also content with that response as to the demographics of the responder, very important information can be obtained in a relatively short time.

It should be noted that the disclosed embodiment is not limited to a single source PC 302, but may encompass a large number of source computers connected over a global communication network. Additionally, the embodiment is not limited to a single ARS 308 or a single advertiser server 312, but may include a plurality of ARS and advertiser systems, indicated by the addition of ARS 314 and advertiser server A 316, respectively. It should also be noted that this embodiment is not limited only to global communication networks, but also may be used with LAN, WAN, and peer-to-peer configurations.

It should also be noted that the disclosed embodiment is not limited to a personal computer, but is also applicable to, for example, a Network Computer ("NetPC"), a scaled-down version of the PC, or any system which accommodates user interaction and interfaces to information resources.

One typical application of the above noted technique is for providing a triggering event during a program, such as a sport event. In a first example, this may be generated by an advertiser. One could imagine that, due to the cost of advertisements in a high profile sports program, there is a desire to utilize this time wisely. If, for example, an advertiser contracted for 15 seconds worth of advertising time, they could insert within their program a tone containing the routing information. This routing information can then be output to the user's PC 302 which will cause the user's PC 302 to, via the network, obtain information from a remote location typically controlled by the advertiser. This could be in the form of an advertisement of a length longer than that contracted for. Further, this could be an interactive type of advertisement. An important aspect to the type of interaction between the actual broadcast program with the embedded routing information and the manufacturer's site is the fact that there is provided information as to the user's PC 302 and a profile of the user themselves. Therefore, an advertiser can actually gain realtime information as to the number of individuals that are watching their particular advertisement and also information as to the background of those individuals, profile information, etc. This can be a very valuable asset to an advertiser.

In another example, the producer of the program, whether it be an on-air program, a program embedded in a video tape, CD-ROM, DVD, or a cassette, can allow the user to automatically access additional information that is not displayed on the screen. For example, in a sporting event, various statistics can be provided to the user from a remote location, merely by the viewer watching the program. When these statistics are provided, the advertiser can be provided with profile information and background information regarding the user. This can be important when, for example, the user may record a sports program. If the manufacturer sees that this program routing code is being output from some device at a time later than the actual broadcast itself, this allows the advertisers to actually

see that their program is still being used and also what type of individual is using it. Alternatively, the broadcaster could determine the same and actually bill the advertiser an additional sum for a later broadcast. This is all due to the fact that the routing information automatically, through a PC and a network, will provide an indication to the advertiser the time at which the actual information was broadcast.

The different type of medium that can be utilized with the above embodiment are such things as advertisements, which are discussed hereinabove, contests, games, news programs, education, coupon promotional programs, demonstration media (demos), and photographs, all of which can be broadcast on a private site or a public site. This all will provide the ability to allow realtime interface with the network and the remote location for obtaining the routed information and also allow for realtime billing and accounting.

Referring now to FIGURE 16, there is illustrated a general block diagram of a disclosed embodiment. A bar code scanning input device 1600 is provided by a input device distributor to customers and is associated with that distributor via a input device ID stored therein. The input device 1600 is either sold or freely distributed to customers for use with their personal computing systems. Since more and more products are being sold using bar codes, it can be appreciated that a user having the input device 1600 can scan bar codes of a multitude of products in order to obtain more information. Information about these products can be made immediately available to the user from the manufacturer for presentation by the user's computer 302. Beyond simply displaying information about the product in which the user is interested, the input device distributor may include additional advertising information for display to the user such as information about other promotions or products provided or sold by the input device distributor. Similarly, advertisers may provide catalogs of advertisements or information in newspapers or periodicals where the user simply scans the bar code associated with the advertisement using the input device 1600 to obtain further information. There is provided a paper source 1602 having contained thereon an advertisement 1604 and an associated bar code 1606. (Note that the disclosed concept is not limited to scanning of bar codes 1606 from paper sources 1602, but is also

operable to scan a bar code 1606 on the product itself. Also, the input device 1600 can be any type of device that will scan any type of image having information encoded therein.)

After obtaining the input device 1600 from the input device distributor, the user connects the input device 1600 to their PC 302. During a scanning operation, input device 1600 reads bar code data 1606 and the input device ID into a "wedge" interface 1608 for conversion into keyboard data, which keyboard data is passed therefrom into the keyboard input port of PC 302. The importance of the input device ID will be discussed in more detail hereinbelow.

The wedge interface 1608 is simply an interface box containing circuitry that accommodates inputs from both the scanning input device 1600 and a computer keyboard 1610. This merely allows the information scanned by the input device 1600 to be input into the PC 302. In the disclosed embodiment, the wedge interface 1608 will convert any information. The data output from the input device 1600 is passed into the wedge interface 1608 for conversion into keyboard data which is readily recognizable by the PC 302. Therefore, the input device 1600 is not required to be connected to a separate port on the PC 302. This data is recognized as a sequence of keystrokes. However, the output of the input device 1600 can be input in any manner compatible with the PC 302. When not receiving scanner data, the wedge interface 1608 simply acts as a pass-through device for keyboard data from the keyboard 1610. In any case, the information is ultimately processed by a processor in the PC 302 and can be presented to the user on a display 1612. The wedge interface 1608 is operable to provide a decoding function for the bar code 1606 and conversion thereof to keystroke input data.

In operation, the product code of a product is provided in the form of a bar code 1606. This bar code 1606 is the "link" to a product. The disclosed embodiment is operable to connect that product information contained in the bar code 1606 with a web page of the manufacturer of that product by utilizing the bar code 1606 as the product "identifier." The program operating on the PC 302 provides routing information to the ARS 308 after launching the browser on the PC 302 and connecting to the ARS 308 over the GCN 306,

which ARS 308 then performs the necessary steps to cause the browser to connect to the manufacturer web site, while also providing for an accounting step, as will be described in more detail hereinbelow.

The bar code 1606 by itself is incompatible with any kind of network for the purposes of communication therewith. It is primarily provided for a retail-type setting. Therefore, the information contained in the bar code 1606, by itself, does not allow for anything other than identification of a product, assuming that one has a database 1614 containing information as to a correlation between the product and the bar code 1606.

The wedge interface 1608 is operable to decode the bar code 1606 to extract the encoded information therein, and append to that decoded bar code information relating to an ID for the input device 1600. This information is then forwarded to the ARS 308 by the resident program in the PC 302. This is facilitated by intermediate routing information stored in the program indicating to which node on the GCN 306 the scanned bar code information is to be sent, i.e., to the ARS 308. It is important to note that the information in the bar code 1606 must be converted from its optical image to numerical values which are then ultimately input to the keyboard input port of PC 302 and converted into data compatible with communication software residing on the PC 302 (in this case, HTML language for insertion into a browser program). When the scanned information is input to the PC 302, the resident program launches the browser program and then assembles a communication packet comprised of the URL of the ARS 308, the input device ID and the user ID. If another type of communications program were utilized, then it would have to be converted into language compatible with that program. Of course, a user could actually key in the information on the bar code 102 and then append the appropriate intermediate routing information thereafter. As will be described hereinbelow, the intermediate routing information appended thereto is the URL of the ARS 308 disposed on the GCN 306.

As part of the configuration for using the input device 1600, the PC 302 hosts input device software which is operable to interpret data transmitted from the input device 1600, and to create a message packet having the scanned product information and input device ID,

routing information, and a user ID which identifies the user location of the input device 1600. The input device software loads at boot-up of the PC 302 and runs in the background. In response to receiving a scanned bar code 1606, the wedge interface 1608 outputs a keystroke code (e.g., ALT-F10) to bring the input device program into the foreground for interaction by the operating system. The input device program then inserts the necessary information into the browser program. The message packet is then transmitted to interface 304 across the global communication network 306 to the ARS 308. The ARS 308 interrogates the message packet and performs a lookup function using the ARS database 310. If a match is found between particular parameters of the message packet, a return message packet is sent back to the PC 302 for processing.

The input device program running on PC 302 functions to partition the browser window displayed to the user into several individual areas. This is for the purpose of preparing to present to the user selected information in each of the individual areas (also called "framing"). The selected information comprises the product information which the user requested by scanning the bar code 1606 using the input device 1600, information about the input device distributor which establishes the identity of the company associated with that particular input device 1600, and at least one or more other frames which may be advertisements related to other products that the input device distributor sells. Note that the advertisements displayed by the input device distributor may be related to the product of interest or totally unrelated. For example, if a user scans the bar code 1606 of a soda from Company A, the input device distributor may generate an advertisement of a new soft drink being marketed by Company A, that it sells. On the other hand, the input device distributor may also structure the display of information to the user such that a user requesting product information of a Product X may get the requested information of Product X along with advertisements for a competing item Product Y. Essentially, the input device distributor is free to generate any advertisement to the user in response to the user requesting product information.

The return message packet transmitted from the ARS 308 to the PC 302 is then transmitted back across the GCN 306 to the advertiser server 312. The advertiser server 312

restructures the message packet and appends the particular product information for transmission back to the PC 302. Upon receiving the particular advertiser information from advertiser server 312, the PC 302 then retransmits a message to the input device distributor site 1616 and E-commerce site 1618 to obtain the information that needs to be framed in the browser window displayed to the user.

Therefore, the input device 1600 is associated with the input device distributor by way of a input device ID such that scanning a product bar code 1606 in order to obtain information about that particular product generates one or more responses from one or more remote sites disposed on the GCN 306. Stored in the input device 1600 is the input device ID which establishes its relationship to the input device distributor. Proprietary input device software running on the PC 302 operates to decode scanned bar code information and the input device ID received from the input device 1600 and wedge interface 1608, and also provides a unique user ID for establishing the location of the user of the input device 1600. The input device software also assembles message packets and works in conjunction with the on-board communication software (e.g., a browser) to automatically route the message packets across the GCN 306 such that the one or more remote sites disposed on the GCN 306 return information to be framed for presentation to the user.

Referring now to FIGURE 17, there is illustrated a conversion circuit of the wedge interface. A microcontroller 1700 provides conversion of the data from the input device 1600 and controls interfacing of the keyboard 1610 and input device 1600 with the PC 302. The microcontroller 1700 has contained therein a memory 1702 or it can have external memory. There are provided a plurality of input device interfaces 1704 to the input device 1600, a plurality of PC interfaces 1706 to the PC 302, and plurality of keyboard interfaces 1708 to the keyboard 1610. In general, the input device interfaces 1704 comprise a serial data line, a ground line, and a power line. Similarly, the keyboard interfaces 1708 comprise a serial data line, a ground line, a clock line, and a power line. The PC 302 provides a clock line, a power line, a serial data, and a ground line for input to the microcontroller 1700. The microcontroller 1700 is operable to receive signals from the keyboard 1610 and transfer the signals to the PC 302 as keyboard signals. Operation with the keyboard 1610 is essentially a

"pass-through" procedure. Data output from the keyboard 1610 is already in keyboard format, and therefore requires no conversion by the wedge interface 1608. With respect to the input device 1600, the serial data is not compatible with a keyboard 1610 and, therefore, it must be converted into a keyboard format in order to allow input thereof to the keyboard input of the PC 302.

The microcontroller 1700 performs this function after decoding this bar code information, and conversion of this bar code information into an appropriate stream of data which is comprised of the bar code information and the appended URL. This appended URL will be pre-stored in the memory 1702 and is programmable at the time of manufacture. It is noted that the memory 1702 is illustrated as being contained within the microcontroller 1702 to provide a single chip solution. However, this could be external memory that is accessible by the microcontroller 1702. Therefore, the microcontroller 1700 provides an interface between the input device 1600 and the keyboard 1610 to the PC 302 which allows the input device 1600 to receive coded information and convert it to keyboard strokes or, alternatively, to merely pass-through the keystrokes from the keyboard 1610. Therefore, the user need not install any type of plug-in circuit board into the motherboard of the PC 302 in order to provide an interface to the input device 1600; rather, the user need only utilize the already available keyboard port in order to input the appropriate data into the system.

In this particular disclosed embodiment, the microcontroller 1700 comprises a PIC16C73 microcontroller by *Microchip Technologies*™. The PIC16C73 device is a low cost CMOS 8-bit microcontroller with an integrated analog-to-digital converter. The PIC16C73 device, as illustrated in the disclosed embodiment, has 192 bytes of RAM and 4k x 4 of EPROM memory. The microcontroller 1700 can accommodate asynchronous or synchronous inputs from input devices connected to it. In this disclosed embodiment, communication to the keyboard 1610 is synchronous while it is asynchronous when communicating with input device 1600.

It should be noted that, although in this particular embodiment bar code information of the bar code 1606 is input into the keyboard input port of the PC 302, disclosed methods may also be advantageously utilized with high speed port architectures such as Universal Serial Bus ("USB") and IEEE 1394.

Bar codes are structured to be read in either direction. Timing considerations need to be addressed because of the variety of individuals scanning the bar code introduce a wide variety of scan rates. Bar codes use bars of varying widths. The presence of a black bar generates a positive pulse, and the absence of a black bar generates no pulse. Each character of a conventional bar code has associated therewith seven pulses or bars. Depending on the width of the bars, the time between pulses varies. In this disclosed embodiment, the interface circuitry 1608 performs a "running" calculation of the scan time based upon the rising edge of the pulses commencing with the leader or header information. The minimum and maximum scans times are calculated continuously in software with the interface 1608 during the scanning process to ensure a successful scan by the user.

Referring now to FIGURE 18, there is illustrated a sample message packet transmitted from the user's PC 302 to the ARS 308. The message packet 1800 comprises a number of bits of information including the bar code information 1802 obtained from the user scanning the bar code 1606 with the input device 1600; the input device ID 1804 which is embedded in a memory in the input device 1600 and identifies it with a particular input device distributor; and a user ID 1806 which is derived from the software running on the PC 302 and which identifies uniquely with the user location. Note that the message packet includes other necessary information for the proper transmission for point to point.

Referring now to FIGURE 19, there is illustrated a more detailed block diagram of the routing of the message packets in order to present the framed information to the user. As is mentioned hereinabove, when the user scans a bar code 1606 using the input device 1600, a input device program running on the user PC 302 is operable to interpret the information output by the input device 1600 and generate a message packet for transmission over the GCN 306. The input device program assembles the message packet such that it is

directed to the ARS 308 disposed on the GCN 306. The message packet contains several pieces of information including the input device ID 1804 which links it to the input device distributor, the user ID 1806 which identifies the particular user using the input device 1600, and bar code information 1802 describing a particular product of interest to the user. This message from the PC 302 is transmitted over a path 1900 to the ARS 308 where the ARS database 310 is accessed to cross reference the ID information 1804 and bar code information 1802 to a particular advertiser and input device distributor. The ARS 308 returns a message packet over a path 1902 to the user PC 302 which contains routing information as to the location of various other sites disposed on the GCN 306, for example, the advertiser server 312 and input device distributor site 1616.

It can be appreciated that other information can also be provided by the ARS 308 which more closely targets the particular user of the input device 1600. For example, if it is known that a particular input device 1600 is sold in a certain geographic area, this information can be useful in targeting the particular user with certain advertising information relevant to that geographic area. In any case, the information returned from the ARS 308 over path 1902 provides enough information for the input device program running on the user PC 302 to identify a number of other sites disposed on the GCN 306. The user PC 302 then processes the return message packet and routes another message packet over a path 1904 to the advertiser server 312. The advertiser server 312 then returns product information of the particular product in which the user was interested back to the user PC 302 over a path 1906. Similarly, the user PC 302 routes information (e.g., the URL of the input device distributor site and the user profile) to the input device distributor site 1616 over a path 1908 in order to obtain information back over a path 1910 for framing any banners which identify the input device distributor. Additionally, the user PC 302 forwards a message packet to the E-commerce site 1618 over a path 1912 in order to return information regarding any particular advertisements the input device distributor wants to display to the user. The advertisements are returned to the PC 302 over a path 1914.

Referring now to FIGURE 20, there is illustrated a block diagram of a browser window according to the disclosed embodiment. The browser window 2000 is partitioned

into a plurality of areas for framing specific information. A bar code area 2002 displays that product information in which the user was interested; an input device-specific area 2004 displays information about the input device distributor; and an E-commerce area 2006 displays advertising information that the input device distributor selects for display according to this particular user and input device 1600. As mentioned hereinabove, a program operable to process scanned bar code information with the unique input device 1600 develops the browser window by partitioning it into specific areas for the framing of information. Therefore, information returned from the E-commerce site 1608 is passed through the GCN 306 to the particular E-commerce frame 2006. Similarly, information about the particular product of interest is returned from the advertiser site 312 across the GCN 306 to the particular bar code specific area 2002. Information placed in the input device specific area 2004 is information about the input device distributor which is returned from the input device distributor site 1616 across GCN 306.

Referring now to FIGURE 21, there is illustrated a structure of information contained in the ARS database. The ARS database 310 contains a variety of information required to properly interrogate and assemble packets for obtaining information from the various sites disposed on the GCN 306. The ARS database 310 has a database structure 2100 which contains addresses for the web sites containing the product information requested by the user when scanning the bar code 1606 with the input device 1600. Under a PRODUCT heading 2102 are listed the particular bar codes and associated routing information for addressing the respective server location. For example, the ARS server 308 may contain any number of advertisers having unique URL addresses associated therewith. Therefore, the bar code 1606 of a particular product is associated with a unique URL address which routes any request for information of that product to that particular advertiser's site. Also part of the ARS database structure 2000 is a heading of INPUT DEVICE under which is the input device ID 1804 and the distributor associated with that input device ID 1804.

It can be appreciated that there may be a number of distributors using the disclosed architecture such that each distributor has an ID embedded in the input device 1600 which

uniquely identifies that input device with the particular distributor. Therefore, the unique input device ID 1804 needs to be listed with the respective distributors of that input device 1600 in order to process the information that needs to be framed and displayed to that particular user. Another heading under the ARS database structure 2100 is a user heading 2106 which contains profile information associated with that particular user ID 1806. As mentioned hereinabove, the user ID 1806 is obtained via the input device software running on the PC 302 and upon installation or subsequent configuration may request that the user input certain profile information which may be used to target that particular user with products and services which identify with that user profile. The ARS database structure 2100 also contains an E-commerce heading 2108 which contains information related to the bar code 1606 and an advertisement that may be triggered by the request for that information. For example, any bar code 1606 associated with a paper source 1602 can be associated with the specific information in the ARS database 310. A user wishing to obtain information about a specific soft drink may, in fact, trigger an advertising response of a competitor product. Similarly, the user interested in information about that particular soft drink may also trigger information which is relevant to that particular product or a product which may normally be served in conjunction with that soft drink. Furthermore, if the user profile indicates that this individual has significant interest in finance or insurance, the request for information regarding this particular bar coded product may trigger advertisement from an E-commerce server 1618 related to information about finance and insurance. It should be noted that the information described as contained within the ARS database structure 2100 is not limited to what has been described, but may comprise any number of pieces of information used to present desired information to the computer display of the user.

Referring now to FIGURE 22, there is illustrated a flowchart of the process of receiving information from the user's perspective, and according to the disclosed embodiment. The input device software running on the user's PC 302 runs in the background until activated by output from the input device 1600. Therefore, flow moves to a decision block 2200 where if a scanned input does not occur, flow moves out the "N" path and loops back to the input of decision block 2200. On the other hand, if scanned input

information is received, flow moves out the "Y" path to a function block 2202 where the input device software assembles a message packet containing the bar code information, the input device ID 1804 and the ARS 308 URL address. Additionally, the browser is launched in which this information is placed for transmission to the ARS 308. Flow then moves to a function block 2204 where the browser is partitioned into any number of areas in which information is displayed when obtained from the input device distributor site 1616, the E-commerce site 1618, and the advertiser server 312. It should be known that although three frames are shown in the particular window 2000 of this embodiment, the number of frames displayed in the window 2000 is limited only by the available real estate of the window 2000 area itself.

After the input device software partitions the browser window into one or more frames in preparation of receipt of return information, flow moves to a decision block 2206 where the computer waits for information to be returned from the various sites disposed on the GCN 306. If information is not returned, flow moves out the "N" path and simply loops back to the input to continue monitoring for receipt of the information. If information has been received, flow moves out the "Y" path to a function block 2208 where routing information for each frame (or partitioned area of the window 2000) is inserted into one or more packets for transmission to the various sites. The various sites then return the requested information back to the PC 302, as indicated in function block 2210. Flow is then to a function block 2212 where the proprietary software working in conjunction with the hosted browser places the returned information into the respective frames of the window. The user, viewing the display at PC 302, then perceives a variety of information, one of which is the particular product information which he or she requested, in addition to input device distributor information, and possibly other advertisements based upon the user's profile.

Referring now to FIGURE 23, there is illustrated a flowchart of the process according to the ARS. The ARS 308 is operable to decode and process messages received from the GCN 306. Therefore, flow is to a decision block 2300 where, if bar code information is not received, flow is out the "N" path with loop-back to its input. If bar code

information has been received, flow is to a function block 2302 where a matching process occurs to link the bar-coded product information to its respective manufacturer. The ARS database 310 also associates the URL address of the manufacturer's server. When a match is found, the ARS 308 begins to assemble a message packet of information for transmission back to the PC 302, as indicated in function block 2304. The message packet contains the product information and the URL address of the manufacturer's website. Flow then moves to a decision block 2306 where the input device ID 1804 is compared with the list of input device IDs issued by the particular input device distributor. If the input device ID 1804 is validated, flow moves out the "Y" path to a function block 2308 where the message packet is appended with the input device ID 1804 and distributor routing address. Flow then moves to a decision block 2310 where the ARS 308 determines if any E-commerce information is to be associated with a particular input device ID 1804. If so, flow is out the "Y" path to a function block 2312 where the message packet is appended with the E-commerce routing string. The E-commerce routing string provides addressing for the E-commerce server 1618. Flow then moves to a function block 2314 where all message packets are returned back to the PC 302 for processing.

Referring back to decision block 2306, if the input device ID 1804 is determined to be invalid, flow moves out the "N" path and jumps forward to the input of decision block 2314, since the lack of a input device ID 1804 interrupts the link to any advertising provided by the E-commerce server 1618. At this point, the only information provided is the link to the advertiser server 312 for return of product information. Referring now to decision block 2310, if no E-commerce information is available, flow moves out the "N" path and jumps forward to the input of function block 2314 where the message packet back to the PC 302 contains only the URL of the advertiser server 312, the bar code information, the distributor server 1616 address and input device ID 1804 information.

Referring now to FIGURE 24, there is illustrated a flowchart of the process performed at the E-commerce site. The E-commerce server 1618 receives the message packet from the user PC 302, as indicated in function block 2400, and decodes the packet to perform a match with the bar coded information. Moving on to a decision block 2402, if the

match is unsuccessful, flow is out the "N" path to a function block 2404 where the match is rejected. A message may be returned to indicate that a problem occurred and the user may need to re-scan the product bar code 1606. If a successful match occurs, flow moves out the "Y" path to a function block 2406 where the input device ID 1804 is matched with the bar code product information. The bar coded information may be distributed to customers over a large geographic area. However, the input device 1606 may be coded for certain geographic areas. For example, a input device 1600 having an XXX ID may be restricted for sale in the Southwestern United States while a input device 1600 having a YYY ID may be sold only in the Northeast. In this way, geographic areas may be targeted with advertising more appealing to that particular area. Advertising returned to the user PC 302 may be focused further by obtaining a user profile when the software or input device 1600 are installed. In this way, advertising may be focused based upon the user profile. Therefore, flow moves to a function block 2408 to lookup the E-commerce action based upon the input device ID 1804 and the bar code information. Flow moves to a function block 2410 to assemble all the information into a packet for return to the user PC 302. The product information and/or user profile information may be returned. Flow is then to a function block 2412 where the message packet is transmitted.

Referring now to FIGURE 25, there is illustrated a general block diagram of another disclosed embodiment relating to a method of accessing one or more remote locations on a network by sensing a machine-resolvable code. The computer (or "PC") 302 is disposed on a network by means of a suitable interface 304. The network may be any type of computer network including a global communications network 306 such as the Internet. The PC 302 is operably connected to an input device 2502 having a sensor 2504 suitable for resolving at least one type of machine-resolvable code 2506. The input device 2502 may be connected directly to the PC 302 or connected through a suitable interface 2507. The interface 2507 may serve more than one device, for examine, FIGURE 25 shows interface 2507 connecting both input device 2502 and keyboard 1610 to the PC 302. Running on the PC 302 is a software application 2520 which includes a software identification code 2522. In response to sensing the machine resolvable code 2506 with the input device 2502, the PC 302 accesses, via the GCN 306, at least one remote location 2524 associated with the software

identification code 2522. The remote locations accessed may then return remote information to the PC 302 for presentation on the display 1612, for storage on the local database 1614, or for other processing.

The machine-resolvable code 2506 may have the form of any type of information which can be sensed and interpreted by a machine, including a bar code 2508, text 2510 suitable for optical character recognition ("OCR"), audio signals 2512 (regardless of whether they are human-audible), optical signals 2514, magnetic signals encoded on a magnetic stripe 2516 such as found on a credit card, or other electromagnetic signals such as those from an induction coupled transceiver device 2518. The machine resolvable code 2506 sensed by the input device 2502 is generally associated with a product, service or event for which information is available from remote sites on the network. For example, the bar code 2508 may be a product bar code, such as a UPC, which is marked directly on a product or package for use in retailing. The bar code could also be of a non-standard type for providing information regarding an advertisement or identifying items displayed in a newspaper periodical or advertisement. The text 2510 may be human readable text which is also machine readable through optical character recognition. The audio signal 2512 may originate from an on-air television or radio program, it may be embedded in the audio track of a video tape, CD Rom, DVD, or cassette. The optical signal 2514 may be a region of modulated brightness on a television screen resulting from a signal received from an on-air television broadcast, or embedded in the video portion of a video tape, CD Rom or DVD. The magnetic information 2516 may be encoded on a credit card or on other types of cards such as those used for identification or access to remote areas. The induction coupled card 2518 could be of the type used for contact-free communication such as between the card and an electromagnetic scanner as might be used in access control systems. The current embodiment utilizes the sensing of the machine-resolvable code 2506, regardless of its exact type and regardless of whether it was originally intended to have any network significance, to cause the computer 302 to access a remote site over the network.

As previously discussed, a software application 2520 runs on the PC 302. The software application 2520 includes a software identification code 2522 associated with at

least one remote site 2524 on the network 306. The associated remote site 2524 (hereinafter the "software ID site") may include information relevant to the software application, or it may include advertising content particularly selected for presentation to users of the software 2520. By use of this embodiment, the computer 302 can access remote sites selected, at least in part, on the basis of an association with the software ID code 2522.

The machine-resolvable code 2506 may be associated with at least a second remote site on the network 306, for example an advertiser server 312. The second remote site may include information relevant to the machine-resolvable code 2506, or it may include advertising content particularly selected for presentation to users in proximity to the machine-resolvable code 2506. By use of further embodiments, the computer 302 can access remote sites selected on the basis of association with the software ID 2522 and the machine-resolvable code 2506.

In operation, a machine-resolvable code ("M-R code") 2506 is sensed by the input device 2502 and then passed to the PC 302. The sensing of the M-R code 2506 may result from an active user activity, such as scanning a bar code or credit card, or it may result from a passive activity at the user's site, such as receiving a TV, radio or other media signal containing an embedded code. In response to the PC 302 receiving the M-R code 2506, the application program 2520 running on the PC converts the software identification code 2522 and generates routing information for transmission to a second computer on the network, e.g., an ARS 308 as previously described.

Referring now FIGURE 26, there is illustrated a sample message packet transmitted from the user's PC 302 to the ARS 308. The message packet 2602 comprises a number of bits of information including the machine-resolvable code 2506 sensed by the input device 2502; the software identification code 2522 which is derived from the software application 2520; and a user ID 1806 which is derived from the software running on the PC 302 and which identifies uniquely with the user's location. Note that the message packet includes other information necessary for the proper transmission of data from point to point across the GCN 306.

Referring now to FIGURE 27, there is illustrated another sample message packet which can be transmitted from the user's PC 302 to the ARS 308. The message packet 2702 comprises a number of bits of information including an input device ID 1804 which is embedded in a memory in the input device 2502 and identifies it with a particular input device distributor, as well as the machine resolvable code 2506, software ID code 2522 and user ID 1806 comprising the message packet previously discussed.

Referring now to FIGURE 28, there is illustrated a more detailed block diagram of the routing of the message packets in order to present the accessed information to the user. As is mentioned hereinabove, when the input device 2502 senses a machine resolvable code 2506, a software program running on the user PC 302 is operable to interpret the information 2802 output by the input device 2502 and generate a message packet for transmission over the GCN 306. The software program assembles the message packet such that it is directed to the ARS 308 disposed on the GCN 306. As seen in FIGUREs 26 and 27, the message packet contains several pieces of information including the software identification code 2522 which associates it with at least one remote site, machine-resolvable code 2506 which associates it with at least a second remote site (e.g., relating to a particular product of interest to the user or to the sponsor of the code), possibly also an input device ID 1804 which links the input device 2502 to the input device distributor and a user ID 1806 which identifies the network location of the user's computer. This message packet from the PC 302 is transmitted over a path 1900 to the ARS 308 where the ARS database 310 is accessed to cross reference the software ID information 2522 to the associated first site (the "software ID site"), the M-R code 2506 to the associated second site (e.g., the advertisers server), and the input device ID 1804 to yet another remote site (e.g., the input device distributor site). The ARS 308 returns a message packet over a path 1902 to the user PC 302 which contains routing information as to the various other sites disposed on the GCN 306, for example, the software ID site 2524, the advertiser server 312, input device distributor site 1616, and the E-commerce site 1618.

It can be appreciated that other information can also be provided by the ARS 308 which more closely targets the particular user sensing the M-R code 2506. For example, if

it is known that software having a particular software code 2522 is sold in a certain geographic area, this information can be useful in targeting the particular user with certain advertising information relevant to that geographical area. In any case, the information returned from the ARS 308 over the path 1902 provides enough information for the input device program running on the user PC 302 to identify a number of remote sites disposed on the GCN 306. The user PC 302 then processes the return message packet and routes another message packet over a path 2810 to the software ID site 2524. The software ID site 2524 then returns information associated with the particular software identification code 2522 which was running on the user PC 302 over a path 2812. Similarly, the user PC 302 routes information (e.g., the URL of the site relating to the M-R code and user profile) to the advertiser server 312 over path 1904 in order to obtain information back over a path 1906 for framing any banners which relate to the sensed M-R code 2506. Additionally, the user PC 302 may forward a message packet to the input device distributor site 1616 over a path 1908 in order to obtain information back over a path 1910 for framing any banners which identify the input device distributor and may also forward a message packet to the E-commerce site 1618 over a path 1912 in order to return information regarding any particular advertisements to be displayed to the user. Such advertisements are returned to the PC 302 over a path 1914.

Referring now to FIGURE 29, there is illustrated a block diagram of a browser window according to another disclosed embodiment. The browser window 2900 is partitioned into a plurality of areas for framing specific information. A M-R code-specific area 2902 displays information relevant to the M-R code 2506, whose sensing activated the user's computer; a software ID-specific area 2904 displays information associated with the software ID 2522 of the software on the user's computer; an input device-specific area 2004 displays information about the input device distributor; and an E-commerce area 2006 displays other advertising information, for example, information that the input device distributor selects for display according to this particular user and input device. As mentioned hereinabove, a software program operable in response to the sensing of M-R code 2506 processes the software ID code 2522 and develops the browser the window by partitioning it into specific areas for the framing of information. Therefore, information

returned from the E-commerce site 1618 is passed through the GCN 306 to the particular E-commerce frame 2006. Similarly, information relevant to the M-R code 2506 is returned from the advertiser site 312 across the GCN 306 to the particular M-R code-specific area 2902. Information placed in the input device-specific area 2004 is information about the input device distributor which is returned from the input device distributor site 1616 across GCN 306. Of course, information placed in the software ID-specific area 2904 is information associated with the software ID 2522 which is returned from the software ID site 2524 across GCN 306.

Referring now to FIGURE 30, there is illustrated another structure of information which may be contained in the ARS database. As previously described, the ARS database 310 contains a variety of information required to properly interrogate and assemble packets for obtaining information from the various remote sites disposed on the GCN 306. The ARS database 310 may have a database structure 3000 which contains addresses or other routing information for the remote sites containing information relevant to codes sent by the user's computer or advertising content specifically selected for presentation to a computer utilizing such codes. Under a SOFTWARE heading 3004, are listed the particular software ID codes and routing information for addressing the associated first remote sites for each respective software ID code. For example, the ARS server 308 may contain any number of advertisers having unique URL addresses associated therewith. Therefore, the software ID code 2522 of a particular software application is associated with a URL address which facilitates the accessing of the desired remote sites based on the software ID code. Also part of the ARS database structure 3000 is a heading M-R CODE 3002 under which is the M-R code whose sensing initiated the action and routing information associated with a second remote site relevant to the respective M-R codes. The ARS database structure 3000 may also contain a USER heading 3006 containing information comprising a user's transaction profile and/or an E-COMMERCE heading 3008 containing codes and routing information for use as previously described.

Referring now to FIGURE 31, there is illustrated a flowchart of the process of receiving information from the user's perspective, and according to another disclosed

embodiment. The software application 2520 running on the user's PC 302 may run in the background until activated by the output from the input device 2502. Therefore, flow moves to a decision block 3100, where if an M-R code has not been sensed, flow moves out the "N" path and loops back to the input of decision block 3100. On the other hand, if an M-R code 2506 has been sensed by the input device 2502, flow moves out the "Y" path to a function block 3102, where the application software 2520 assembles a message packet containing the M-R code 2506, the software ID code 2522 and the URL address for the ARS 308. Additionally, a browser is launched in which this information is placed for transmission to the ARS 308. Flow then moves to a function block 3104 where the browser partitions the screen into any number of areas ("frames") in which information is displayed when obtained from the various remote sites, for example, from the advertiser server 312 and from the software ID site 2524. It should be known that although four frames are shown in the particular window 2900 of this embodiment, the number of frames displayed in the window 2900 may be greater or less depending upon the desired effect.

After the software application partitions the browser window into one or more frames in preparation for receipt of return information, flow moves to a decision block 3106 where the computer waits for information to be returned from the ARS 308 disposed on the GCN 306. If the information is not returned, flow moves out the "N" path and simply loops back to the input to continue monitoring for receipt of the information. If information has been received, flows moves out the "Y" path to a function block 3108 where routing information corresponding to each frame (or partitioned area of the window 2900) is inserted into one or more packets for transmission to the various sites on the GCN 306. The various sites then return the requested information back to the PC 302, as indicated in function block 3110. Flow is then to a function block 3112, where the software application 2520 working in conjunction with the browser places the returned information into the respective frames of the window 2900. The user, viewing the display at PC 302, then perceives a variety of information, at least a portion of which is information associated with the software ID 2522, and possibly other information or advertisements based upon the M-R code 2506, the input device ID 1804 and possibly other advertisements based on the user's profile.

Referring now to FIGURE 32, there is illustrated a flowchart of the process of the current embodiment according to the ARS. The ARS 308 is operable to decode and process messages received from the GCN 306. Therefore, flow is to a decision block 3200 where, if a message is not received, flow is out the "N" path with loop back to its input. If a message has been received, flow is to a function block 3202 where the message is analyzed to see if a software ID code 2522 is included. If no software ID code received, flow is out the "N" path to decision block 3208 for further processing as described below. If a software ID code is included in the message, flow is to a function block 3204 where the ARS database 310 is consulted to match the software ID code 2522 to at least one associated remote site 2524. Flow then moves to function blocks 3206 where the routing string for the software ID site 2524 is assembled. Flow then continues to a decision block 3208 where the message is analyzed to see if additional codes have been received, for example, M-R codes 2506, or input device ID code 1804. If such additional codes are present, the ARS database 310 is again consulted to match the additional codes to associated remote sites and assemble the corresponding routing strings. Once these additional codes have been assembled, or if the output from decision block 3208 follows the "N" path, flow continues to function block 3212 where all routing strings are returned from the ARS 308 to the PC 302 using information from the user ID 1806.

Referring now to FIGURE 33, there is illustrated a flowchart of the process performed at the software ID site. The software ID server 2524 receives the message packet from the user PC 302, as indicated in the function block 3300. This message packet was routed to the software ID server 2524 using the routing string information returned from the ARS 308. Moving on to a function block 3302, the decoded message packet is used to assemble information from the server 2524 which is associated with the software ID code 2522 into a new message packet including information for routing the package back to the user's PC 302. Flow then continues to function block 3304 where the software ID server 2524 transmits the message packet including the information associated with the software ID 2522 back to the user PC 302.

Although the preferred embodiment has been described in detail, it should be understood that various changes, substitutions and alterations can be made therein without departing from the spirit and scope of the invention as defined by the appended claims.

WHAT IS CLAIMED IS:

1. A method of accessing one or more remote locations on a network by sensing a machine-resolvable code, comprising the steps of:

providing a first computer disposed on the network, the first computer being connected to an input device for sensing a machine-resolvable code present at a first location, the first computer running a software application which includes a software identification code having associated therewith at least one of the one or more remote locations;

accessing with the first computer a second computer disposed on the network in response to sensing the machine-resolvable code present at the first location with the input device;

performing a lookup operation at the second computer to match the software identification code with the associated at least one of the one or more remote locations to obtain remote routing information corresponding to the associated at least one of the one or more remote locations;

returning the remote routing information corresponding to the associated at least one of the one or more remote locations from the second computer to the first computer; and

accessing with the first computer the associated at least one of the one or more remote locations according to the remote routing information to retrieve remote information.

2. The method of Claim 1, wherein the step of accessing further comprises the steps of:

returning information from the associated at least one of the one or more remote locations to the first computer; and

presenting at least a portion of the information so returned on the display of the first computer for presentation to the user.

3. The method of Claim 1, wherein in response to the sensing of a machine-resolvable code using the input device, the software application running on the first computer converts the software identification code and generates routing information for transmission to the second computer.

4. The method of Claim 3, wherein the routing information includes the software identification code and the address of the second computer.

5. The method of Claim 1, wherein the machine-resolvable code is an optical code and the input device is an optical code scanner.

6. The method of Claim 5, wherein the optical code is a bar code and the optical code scanner is a bar code scanner.

7. The method of Claim 6, wherein the bar code is a universal product code (UPC) bar code.

8. The method of Claim 5, wherein the optical code is alphanumeric text and the optical code scanner is an optical character recognition (OCR) scanner.

9. The method of Claim 5, wherein the optical code is a portion of a display screen displaying a pattern of modulated brightness and the optical code scanner comprises a light sensor.

10. The method of Claim 1, wherein the machine-resolvable code is an audio tone and the input device comprises a microphone.

11. The method of Claim 1, wherein the machine-resolvable code is a magnetic pattern in a strip of magnetic material and the input device is a magnetic strip reader.

12. The method of Claim 1, wherein the machine-resolvable code is a pattern of electromagnetic signals transmitted from an induction-coupled transceiver device and the input device is an electromagnetic signal receiver.

13. The method of Claim 1, wherein the machine-resolvable code is associated with at least a second of the one or more remote locations and the step of performing a lookup operation at the second computer further comprises matching the machine-resolvable code with the associated at least a second of the one or more remote locations to obtain remote routing information corresponding to the associated at least a second of the one or more remote locations.

14. The method of Claim 13, wherein the step of returning the remote routing information further comprises returning the remote routing information corresponding to the associated at least a second of the one or more remote locations from the second computer to the first computer.

15. The method of Claim 14, wherein the step of accessing further comprises the steps of,

returning information from the associated at least one of the one or more remote locations to the first computer;

returning information from the associated second of the one or more remote locations to the first computer; and

framing at least a portion of the information from the associated at least one of the one or more remote locations and at least a portion of the information from the associated second of the one or more remote locations in a browser window of the first computer for presentation to the user.

16. The method of Claim 1,wherein the network is a global communication network.

17. A system for accessing one or more remote locations on a network by sensing a machine-resolvable code, comprising:

a first computer disposed on the network, the first computer being connected to an input device for sensing a machine-resolvable code present at a first location, the first computer running a software application which includes a software identification code having associated therewith at least one of the one or more remote locations;

a second computer disposed on the network, and accessed in response to the input device sensing the machine-resolvable code present at the first location;

wherein a lookup operation is performed at the second computer to match the software identification code with the associated at least one of the one or more remote locations to obtain remote routing information corresponding to the associated at least one of the one or more remote locations;

wherein the remote routing information corresponding to the associated at least one of the one or more remote locations is returned from the second computer to the first computer; and

wherein the associated at least one of the one or more remote locations are accessed by the first computer according to the remote routing information to retrieve remote information therefrom.

18. The system of Claim 17, wherein at least a portion of the information returned from the associated at least one of the one or more remote locations to the first computer is presented on the display of the first computer.

19. The system of Claim 17, wherein the software application running on the first computer converts the software identification code and generates routing information for transmission to the second computer in response to the sensing of a machine-resolvable code by the input device.

20. The system of Claim 19, wherein the routing information includes the software identification code and the address of the second computer.

21. The system of Claim 17, wherein the machine-resolvable code is an optical code and the input device is an optical code scanner.

22. The system of Claim 21, wherein the optical code is a bar code and the optical code scanner is a bar code scanner.

23. The system of Claim 22, wherein the bar code is a universal product code (UPC) bar code.

24. The system of Claim 21, wherein the optical code is alphanumeric text and the optical code scanner is an optical character recognition (OCR) scanner.

25. The system of Claim 21, wherein the optical code is a portion of a display screen displaying a pattern of modulated brightness and the optical code scanner comprises a light sensor.

26. The system of Claim 17, wherein the machine-resolvable code is an audio tone and the input device comprises a microphone.

27. The system of Claim 17, wherein the machine-resolvable code is a magnetic pattern in a strip of magnetic material and the input device is a magnetic strip reader.

28. The system of Claim 17, wherein the machine-resolvable code is a pattern of electromagnetic signals transmitted from an induction-coupled transceiver device and the input device is an electromagnetic signal receiver.

29. The system of Claim 17, wherein the machine-resolvable code is associated with at least a second of the one or more remote locations; and wherein the second computer performs a lookup operation matching the machine-resolvable code with the associated at

least a second of the one or more remote locations to obtain remote routing information corresponding to the associated at least a second of the one or more remote locations.

30. The method of Claim 29, wherein the second computer returns the remote routing information corresponding to the associated at least a second of the one or more remote locations to the first computer.

31. The method of Claim 30, wherein information from the associated at least one of the one or more remote locations is returned to the first computer; wherein information from the associated second of the one or more remote locations is returned to the first computer; and wherein at least a portion of the information from the associated at least one of the one or more remote locations and at least a portion of the information from the associated second of the one or more remote locations are framed in a browser window of the first computer for presentation to the user.

32. The system of Claim 17,wherein the network is a global communication network.

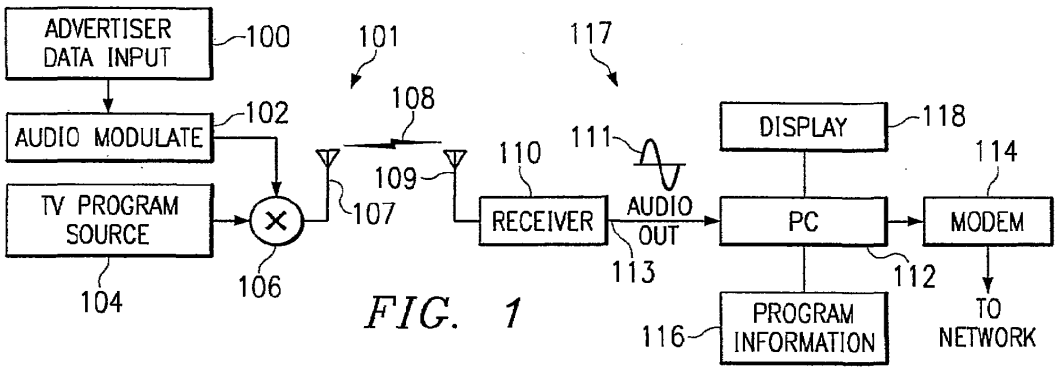


FIG. 1

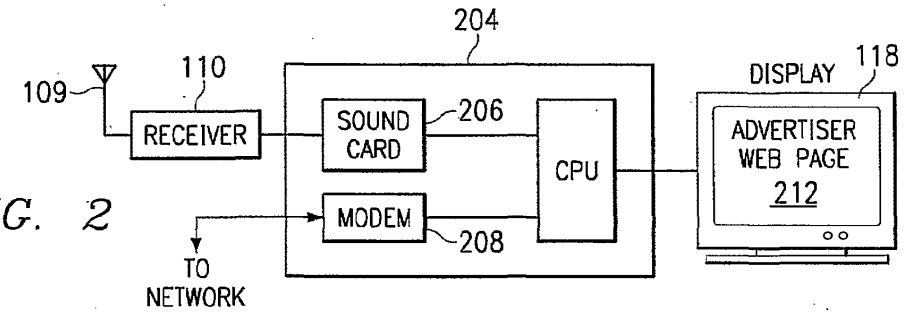


FIG. 2

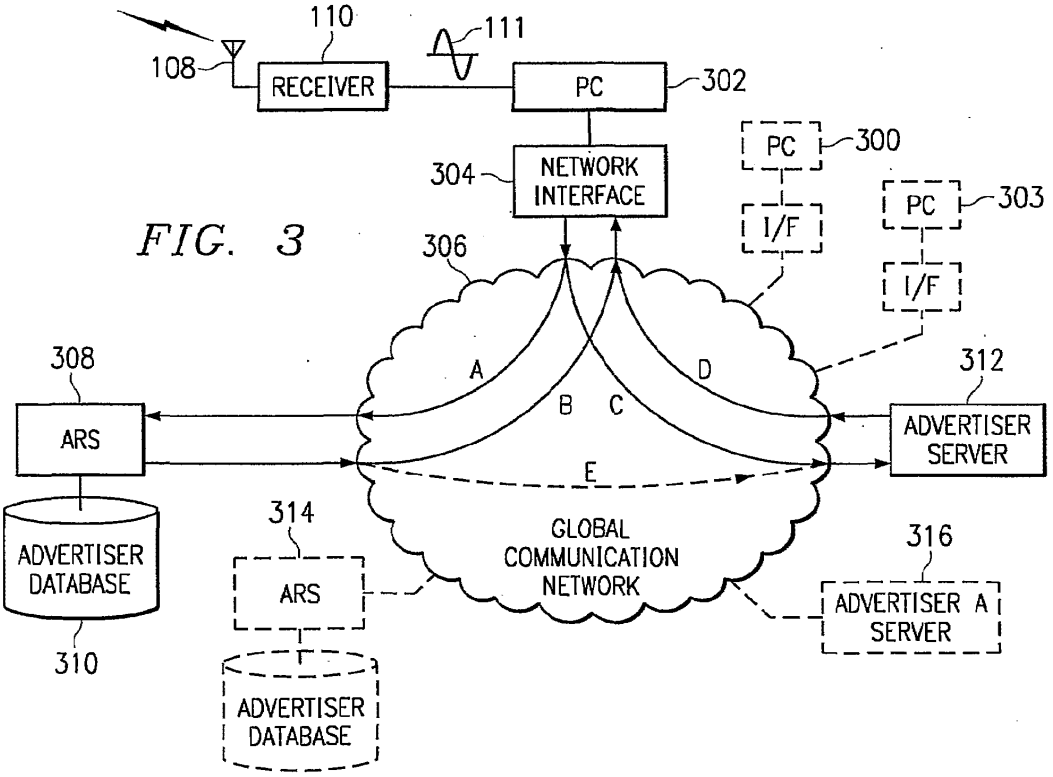


FIG. 3

PATH A: SOURCE TO ARS

400	URL OF ARS	ADVERTISER PRODUCT CODE	URL OF SOURCE

FIG. 4a

PATH B: ARS TO SOURCE

402	URL OF SOURCE	URL OF ADVERTISER	URL OF ARS

FIG. 4b

PATH C: SOURCE TO ADVERTISER

404	URL OF ADVERTISER	REQUEST-FOR-INFORMATION	URL OF SOURCE

FIG. 4c

PATH D: ADVERTISER TO SOURCE

406	URL OF SOURCE	REQUESTED INFORMATION	URL OF ADVERTISER

FIG. 4d

PATH E: ARS TO ADVERTISER (OPTIONAL)

408	URL OF ADVERTISER	REQUEST-FOR-INFORMATION	URL OF SOURCE

FIG. 4e

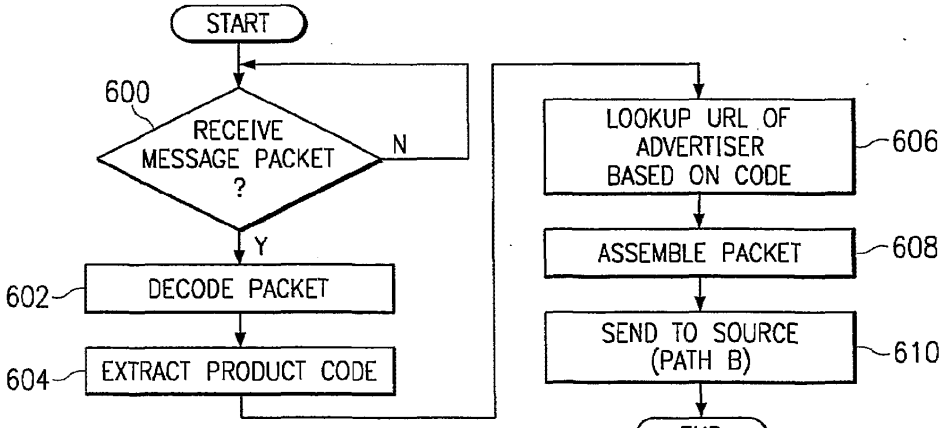


FIG. 6

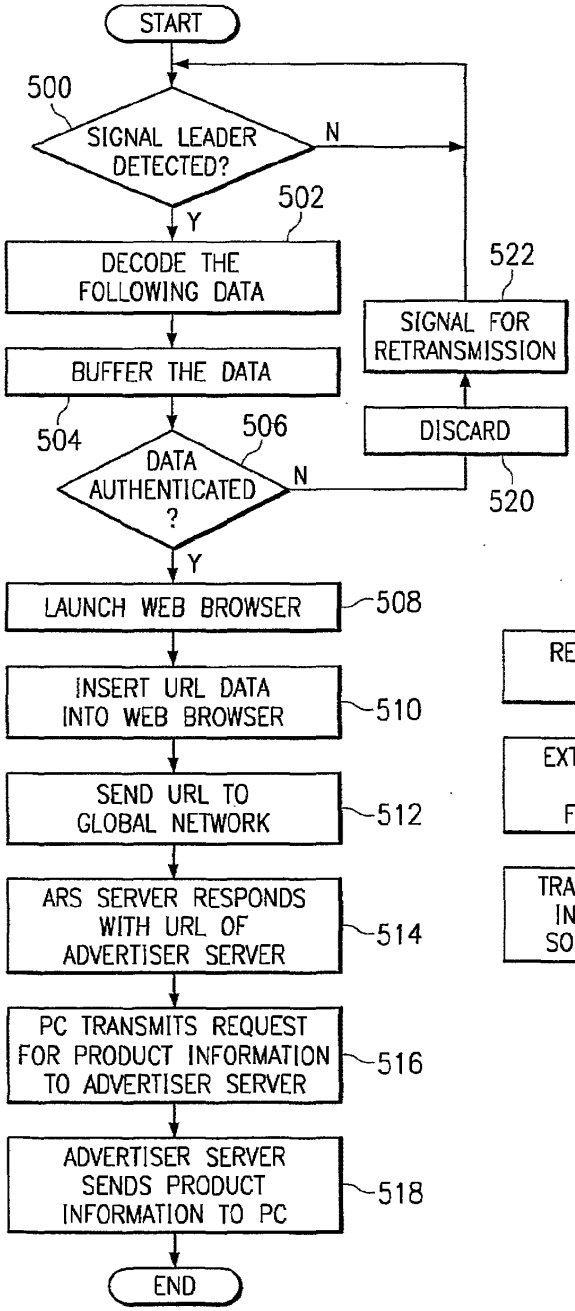

START
500
SIGNAL LEADER DETECTED?
N
Y
502
DECODE THE FOLLOWING DATA
BUFFER THE DATA
504
506
DATA AUTHENTICATED ?
N
Y
522
SIGNAL FOR RETRANSMISSION
DISCARD
520
LAUNCH WEB BROWSER
508
INSERT URL DATA INTO WEB BROWSER
510
SEND URL TO GLOBAL NETWORK
512
ARS SERVER RESPONDS WITH URL OF ADVERTISER SERVER
514
PC TRANSMITS REQUEST FOR PRODUCT INFORMATION TO ADVERTISER SERVER
516
ADVERTISER SERVER SENDS PRODUCT INFORMATION TO PC
518
END

FIG. 5

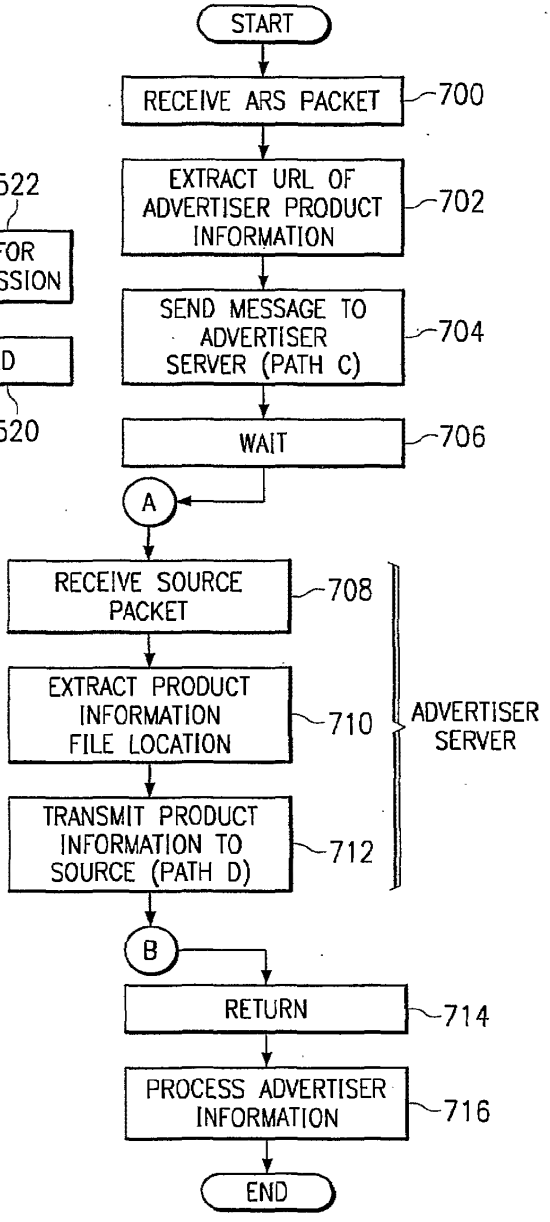

START
RECEIVE ARS PACKET
700
EXTRACT URL OF ADVERTISER PRODUCT INFORMATION
702
SEND MESSAGE TO ADVERTISER SERVER (PATH C)
704
WAIT
706
A
RECEIVE SOURCE PACKET
708
EXTRACT PRODUCT INFORMATION FILE LOCATION
710
TRANSMIT PRODUCT INFORMATION TO SOURCE (PATH D)
712
ADVERTISER SERVER
B
RETURN
714
PROCESS ADVERTISER INFORMATION
716
END

FIG. 7

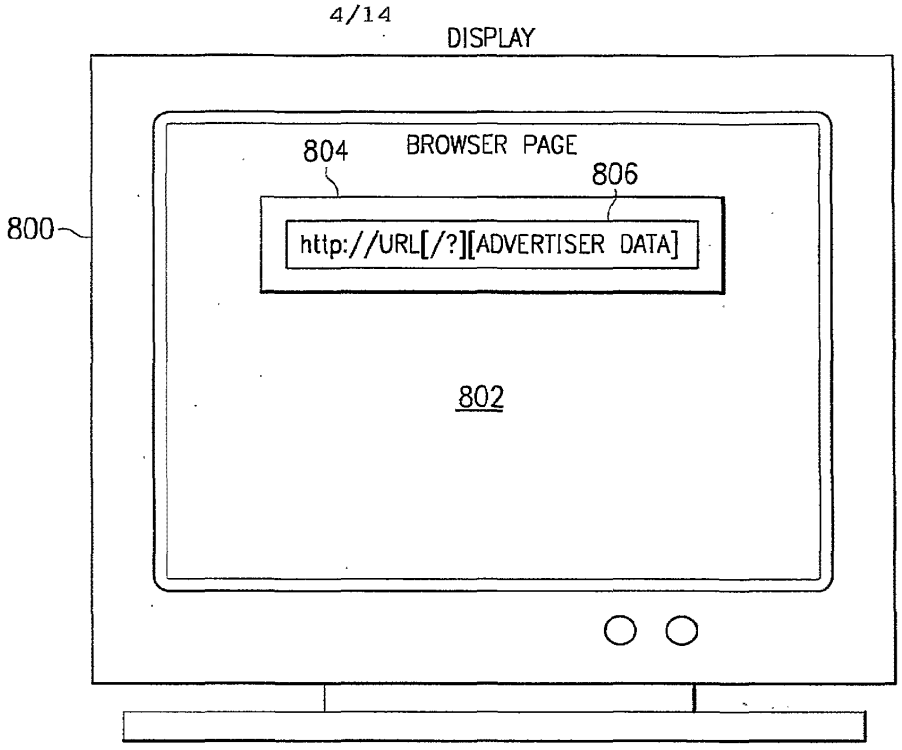
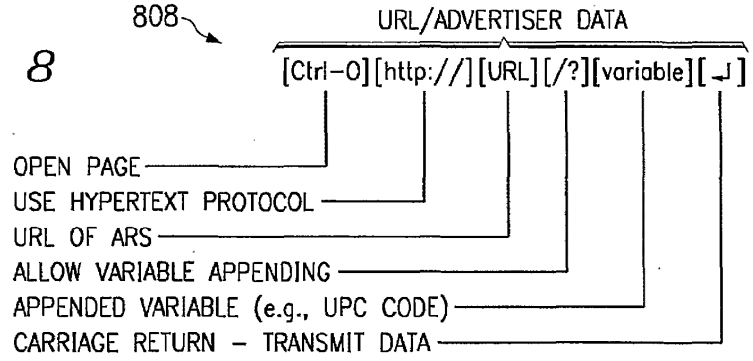

DISPLAY
800
804
BROWSER PAGE
806
http://URL[/?][ADVERTISER DATA]
802

FIG. 8


808
URL/ADVERTISER DATA
[Ctrl-O][http://][URL][/?][variable][↵]
OPEN PAGE
USE HYPERTEXT PROTOCOL
URL OF ARS
ALLOW VARIABLE APPENDING
APPENDED VARIABLE (e.g., UPC CODE)
CARRIAGE RETURN - TRANSMIT DATA

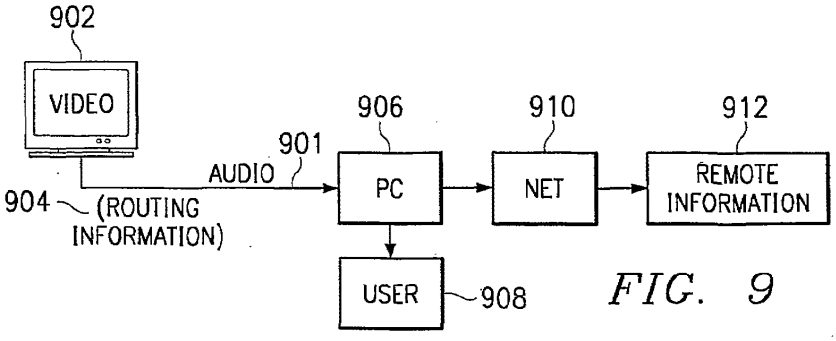

902
VIDEO
906
910
912
901
AUDIO
PC
NET
REMOTE INFORMATION
904
(ROUTING INFORMATION)
USER
908

FIG. 9

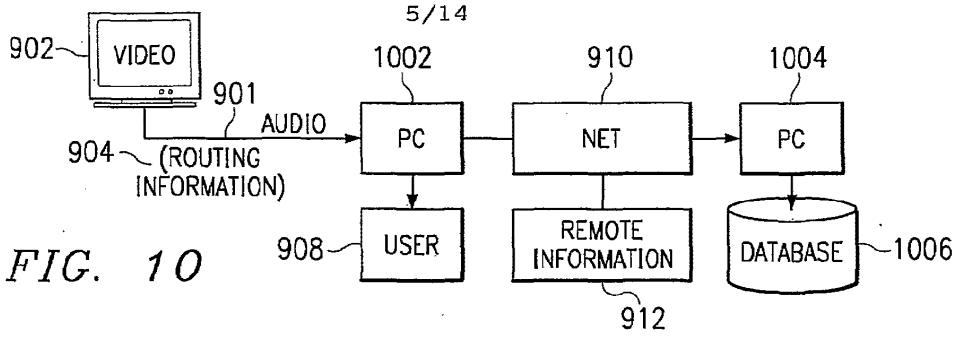


FIG. 10

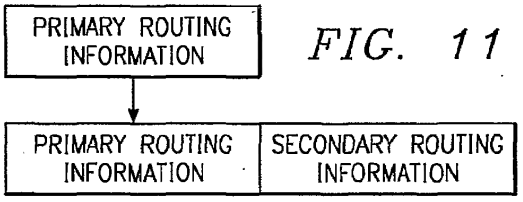


FIG. 11

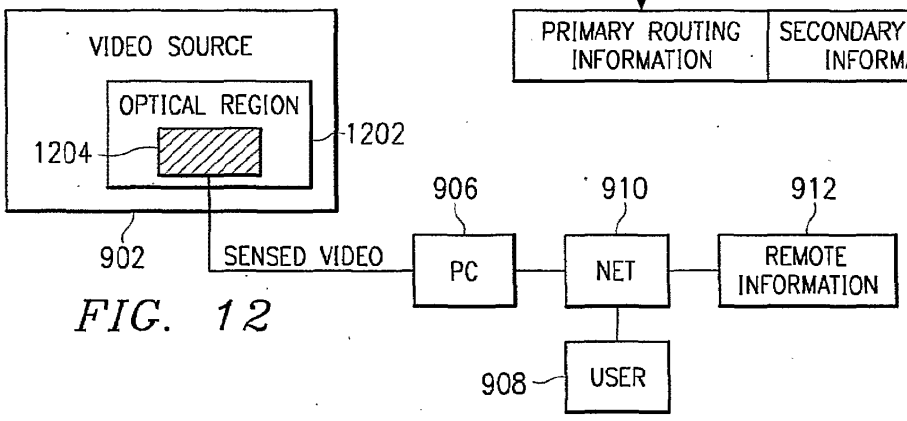


FIG. 12

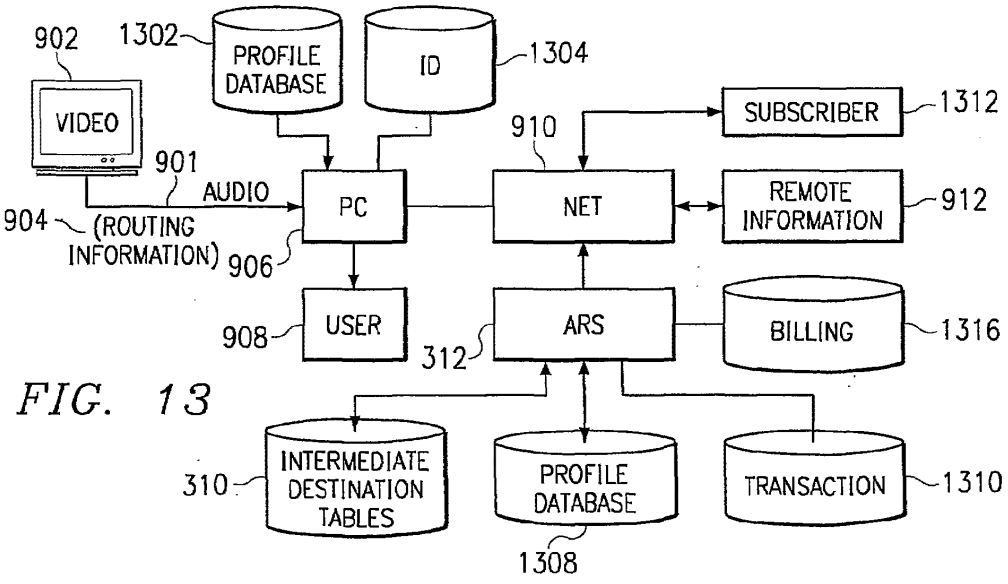


FIG. 13

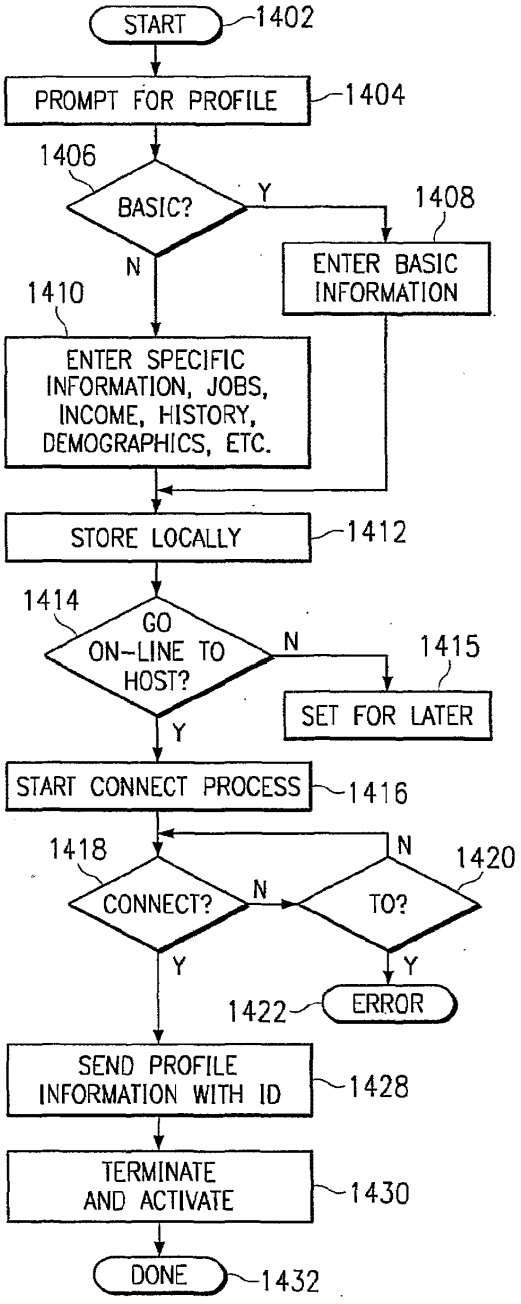

START
1402
PROMPT FOR PROFILE
1404
1406
BASIC?
Y
1408
ENTER BASIC INFORMATION
N
1410
ENTER SPECIFIC INFORMATION, JOBS, INCOME, HISTORY, DEMOGRAPHICS, ETC.
STORE LOCALLY
1412
1414
GO ON-LINE TO HOST?
N
1415
SET FOR LATER
Y
START CONNECT PROCESS
1416
1418
CONNECT?
N
N
1420
TO?
Y
Y
1422
ERROR
SEND PROFILE INFORMATION WITH ID
1428
TERMINATE AND ACTIVATE
1430
DONE
1432

FIG. 14

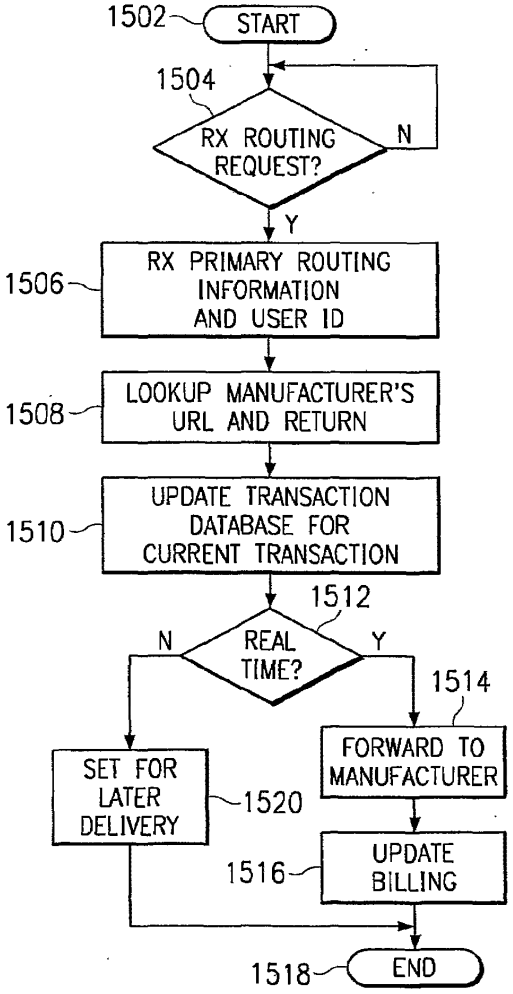

1502
START
1504
RX ROUTING REQUEST?
N
Y
1506
RX PRIMARY ROUTING INFORMATION AND USER ID
1508
LOOKUP MANUFACTURER'S URL AND RETURN
1510
UPDATE TRANSACTION DATABASE FOR CURRENT TRANSACTION
1512
N
REAL TIME?
Y
1514
FORWARD TO MANUFACTURER
SET FOR LATER DELIVERY
1520
1516
UPDATE BILLING
1518
END

FIG. 15

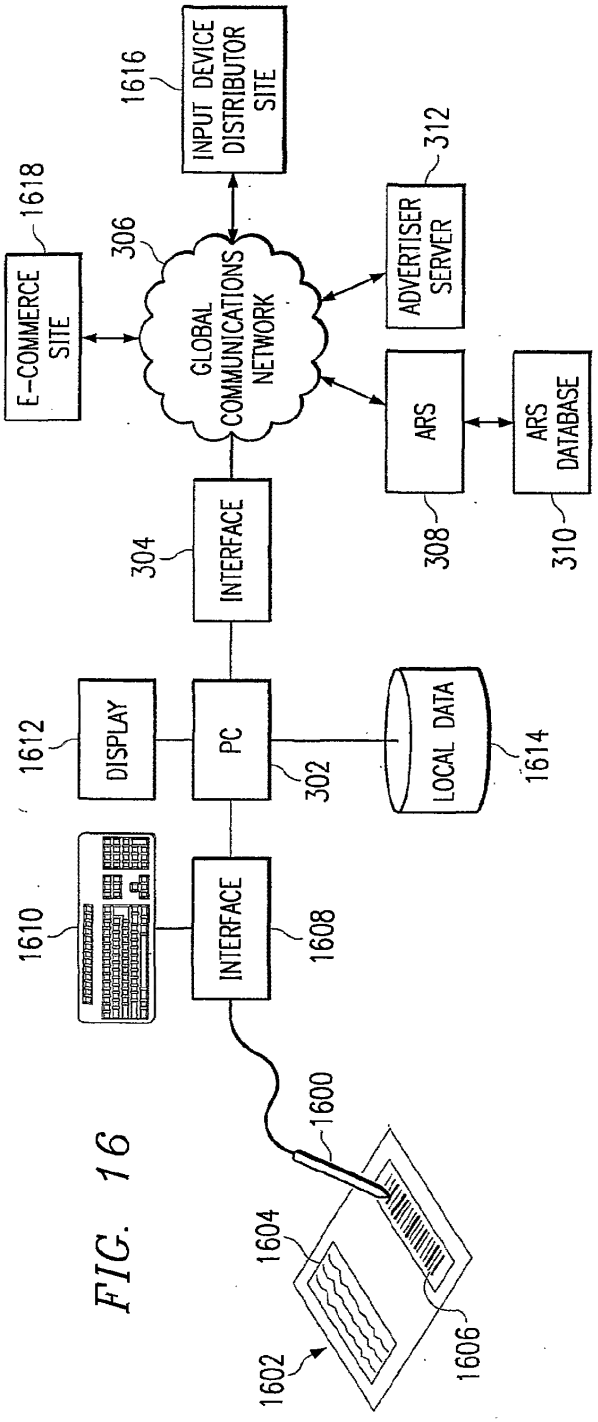


FIG. 16

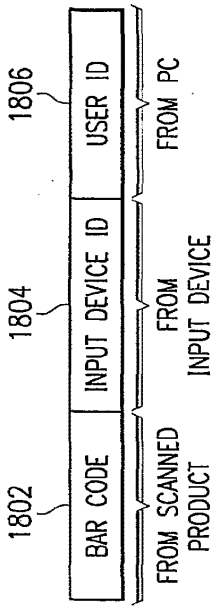


FIG. 18

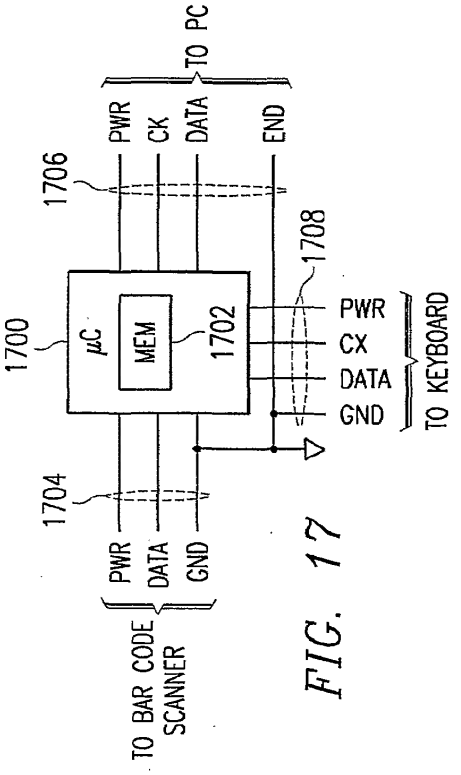


FIG. 17

FIG. 19

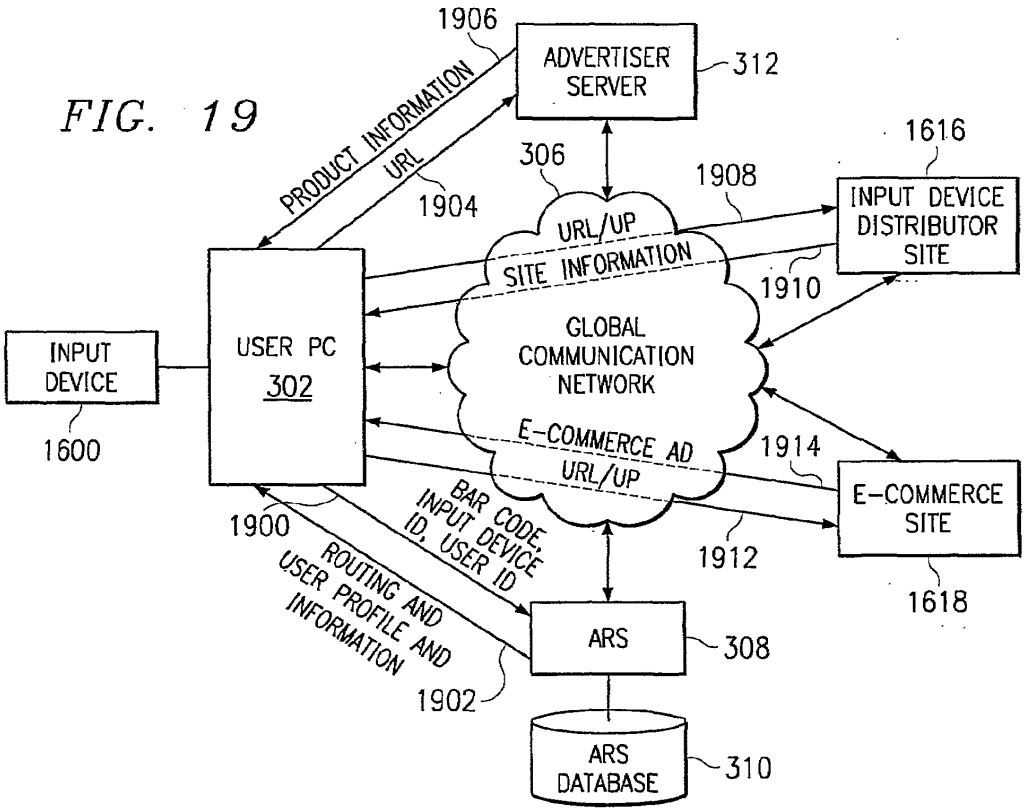

1906
ADVERTISER SERVER
312
PRODUCT INFORMATION
URL
1904
306
1908
1616
INPUT DEVICE DISTRIBUTOR SITE
URL/UP
SITE INFORMATION
1910
INPUT DEVICE
1600
USER PC
302
GLOBAL COMMUNICATION NETWORK
E-COMMERCE AD
URL/UP
1914
E-COMMERCE SITE
1618
1900
BAR CODE, INPUT DEVICE ID, USER ID
ROUTING AND USER PROFILE AND INFORMATION
1912
ARS
308
1902
ARS DATABASE
310

FIG. 20

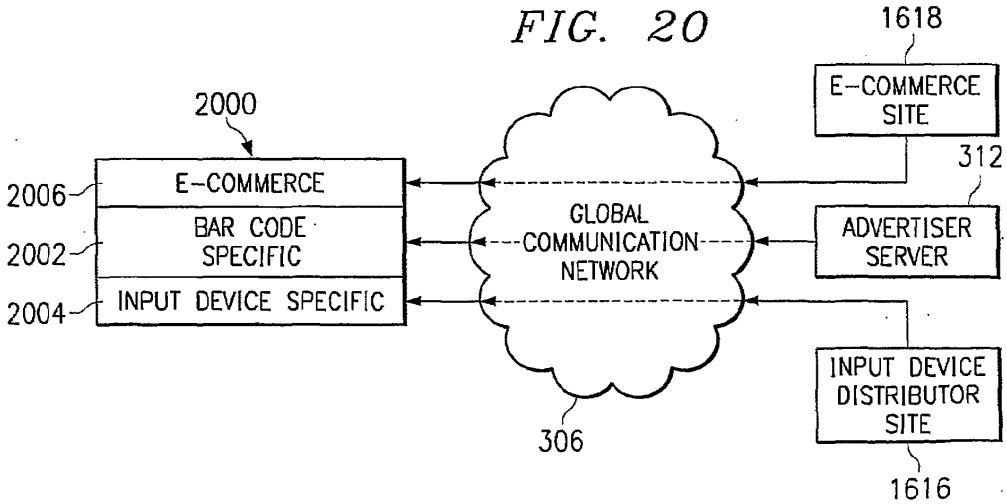

1618
E-COMMERCE SITE
2000
312
2006
E-COMMERCE
2002
BAR CODE SPECIFIC
2004
INPUT DEVICE SPECIFIC
GLOBAL COMMUNICATION NETWORK
ADVERTISER SERVER
INPUT DEVICE DISTRIBUTOR SITE
306
1616

FIG. 21

ARS DATABASE 2100

2102		2104		2106	2108	
PRODUCT		INPUT DEVICE		USER	E-COMMERCE	
BAR CODE	ROUTE	ID	DISTRIBUTOR	PROFILE	BAR CODE	INFORMATION

FIG. 22

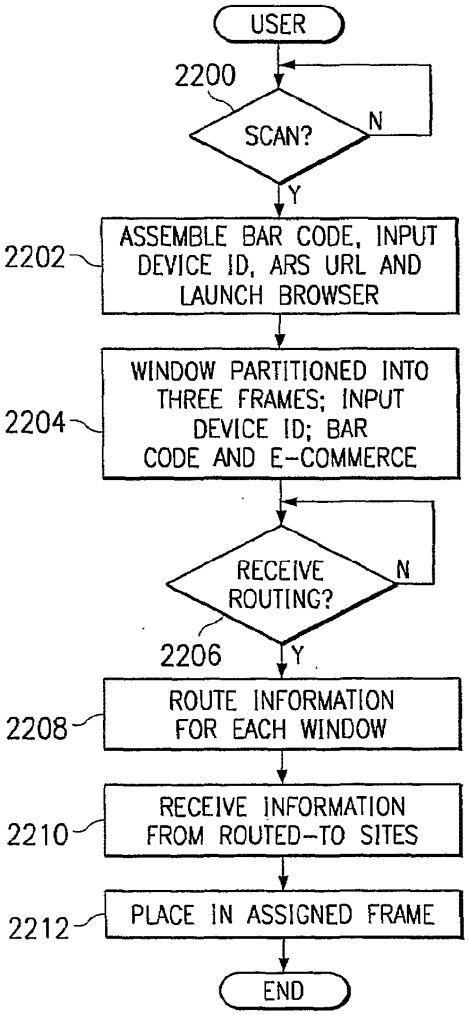


FIG. 23

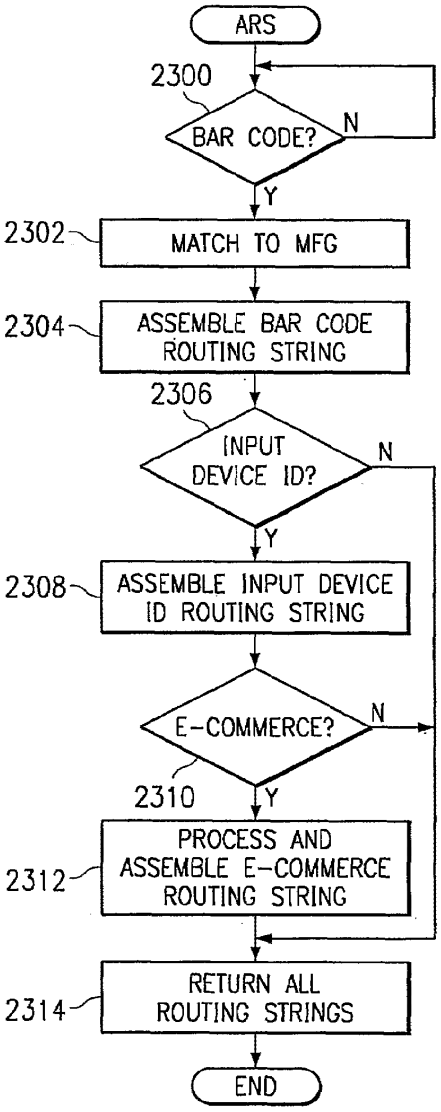
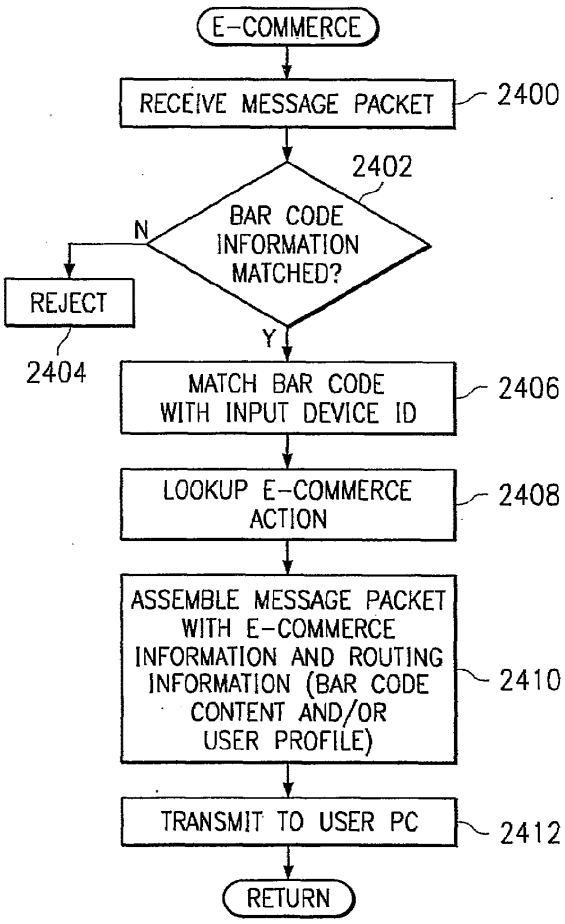

ARS
2300
BAR CODE?
N
Y
2302
MATCH TO MFG
2304
ASSEMBLE BAR CODE ROUTING STRING
2306
INPUT DEVICE ID?
N
Y
2308
ASSEMBLE INPUT DEVICE ID ROUTING STRING
E-COMMERCE?
N
2310
Y
2312
PROCESS AND ASSEMBLE E-COMMERCE ROUTING STRING
2314
RETURN ALL ROUTING STRINGS
END

FIG. 24


E-COMMERCE
RECEIVE MESSAGE PACKET
2400
2402
BAR CODE INFORMATION MATCHED?
N
REJECT
2404
Y
MATCH BAR CODE WITH INPUT DEVICE ID
2406
LOOKUP E-COMMERCE ACTION
2408
ASSEMBLE MESSAGE PACKET WITH E-COMMERCE INFORMATION AND ROUTING INFORMATION (BAR CODE CONTENT AND/OR USER PROFILE)
2410
TRANSMIT TO USER PC
2412
RETURN

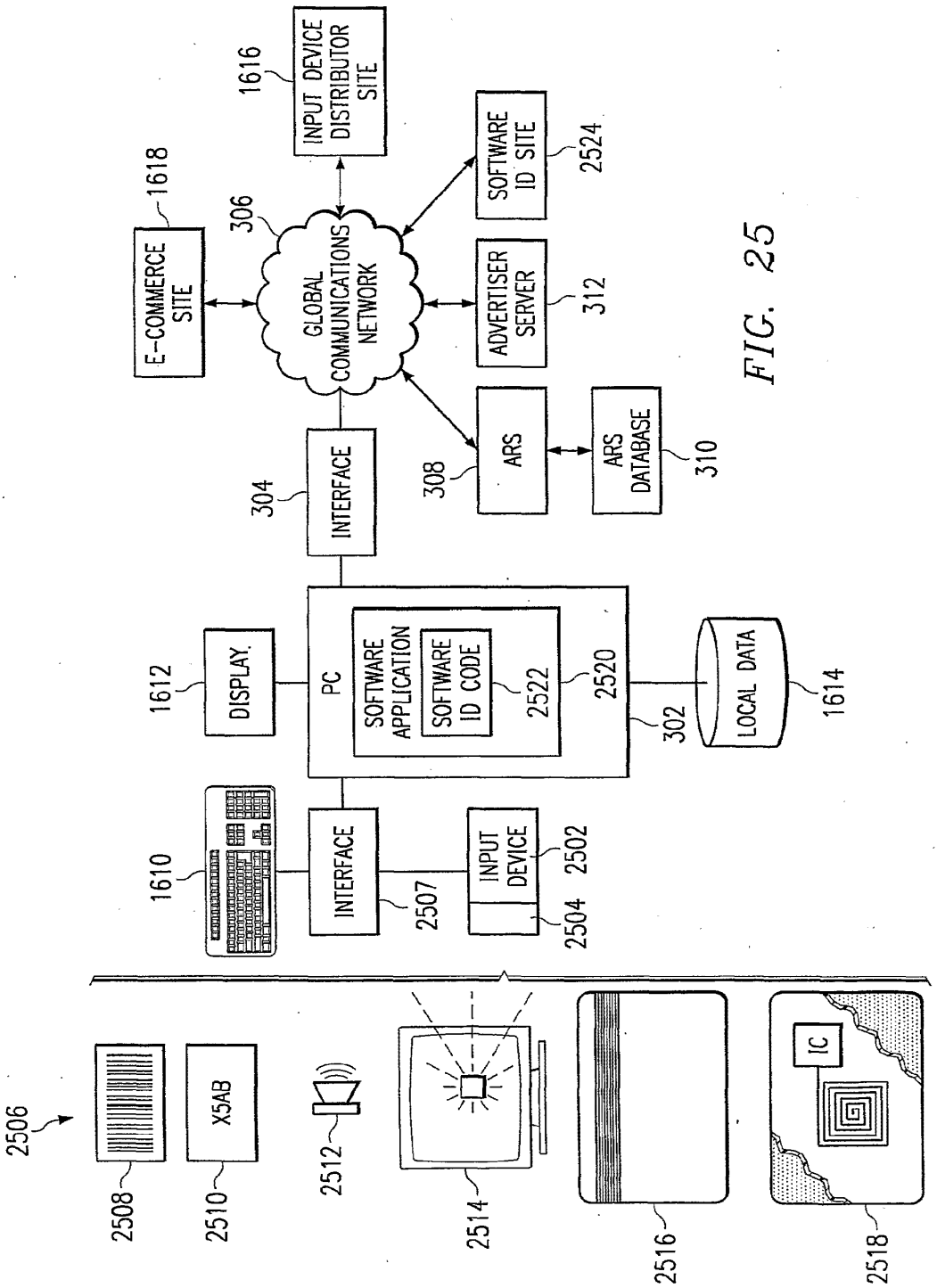

INPUT DEVICE DISTRIBUTOR SITE
1616
SOFTWARE ID SITE
2524
1618
306
E-COMMERCE SITE
GLOBAL COMMUNICATIONS NETWORK
ADVERTISER SERVER
312
INTERFACE
304
308
ARS
ARS DATABASE
310
DISPLAY
1612
PC
SOFTWARE APPLICATION
SOFTWARE ID CODE
2522
2520
302
LOCAL DATA
1614
1610
INTERFACE
2507
INPUT DEVICE
2504
2502
2506
2508
X5AB
2510
2512
2514
2516
IC
2518

FIG. 25

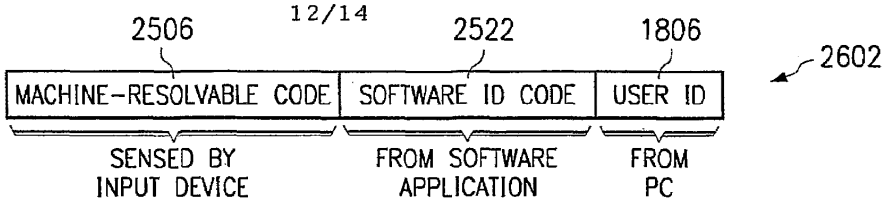


FIG. 26

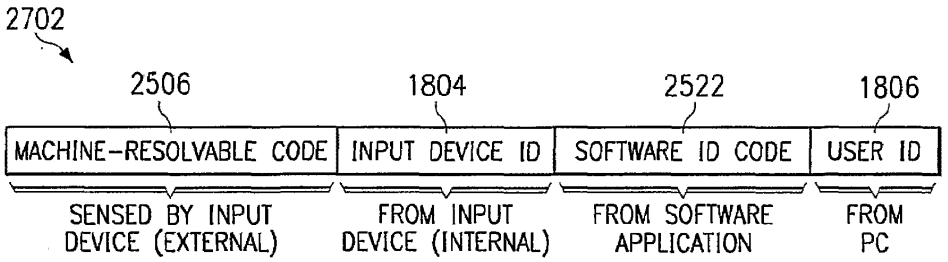


FIG. 27

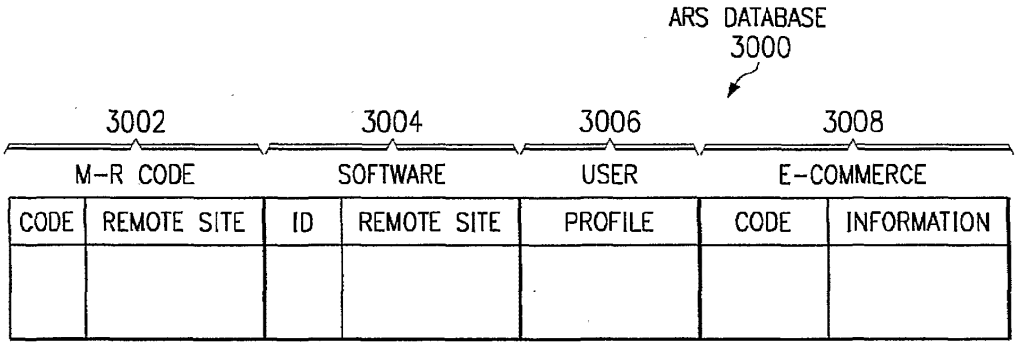


FIG. 30

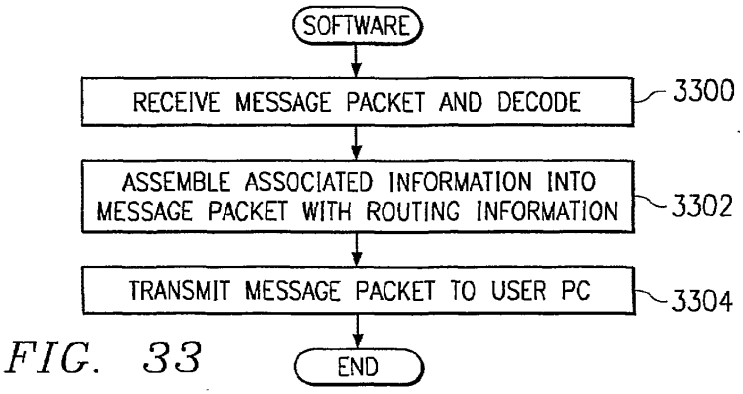


FIG. 33

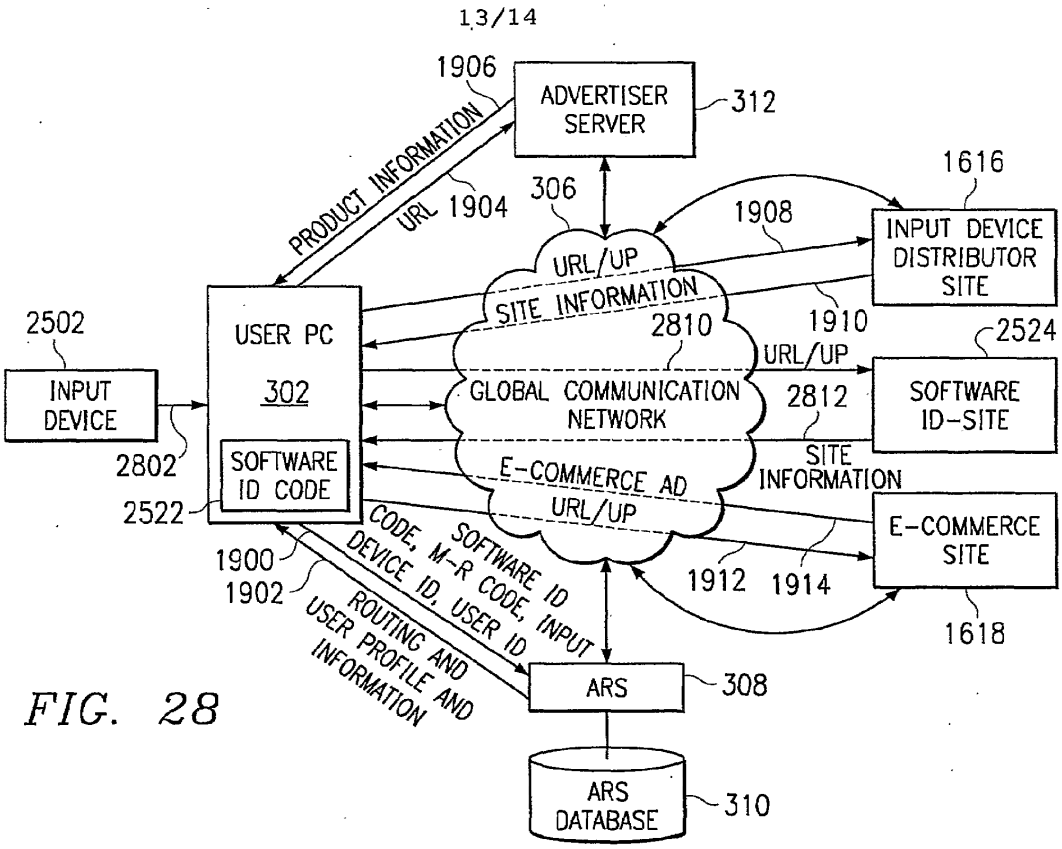

1906
ADVERTISER SERVER
312
PRODUCT INFORMATION
URL 1904
306
1616
1908
INPUT DEVICE DISTRIBUTOR SITE
URL/UP
SITE INFORMATION
2810
1910
2524
2502
USER PC
302
URL/UP
INPUT DEVICE
GLOBAL COMMUNICATION NETWORK
2812
SOFTWARE ID-SITE
2802
SOFTWARE ID CODE
SITE INFORMATION
E-COMMERCE AD
2522
URL/UP
E-COMMERCE SITE
1900
1902
CODE, SOFTWARE ID
DEVICE ID, USER ID
M-R CODE, INPUT
1912
1914
ROUTING AND USER PROFILE AND INFORMATION
1618
ARS
308
ARS DATABASE
310

FIG. 28

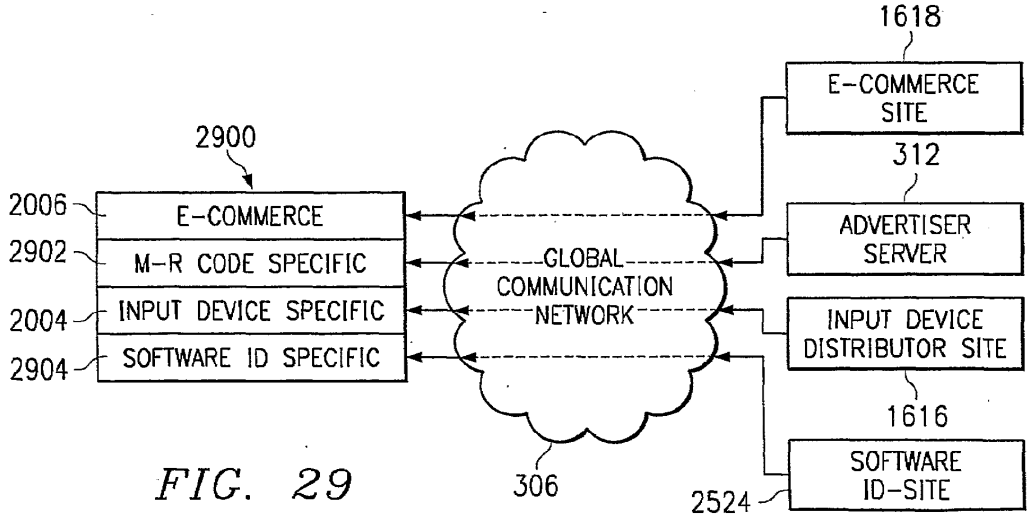

1618
E-COMMERCE SITE
2900
312
2006
E-COMMERCE
ADVERTISER SERVER
2902
M-R CODE SPECIFIC
GLOBAL COMMUNICATION NETWORK
INPUT DEVICE DISTRIBUTOR SITE
2004
INPUT DEVICE SPECIFIC
1616
2904
SOFTWARE ID SPECIFIC
SOFTWARE ID-SITE
306
2524

FIG. 29

FIG. 31

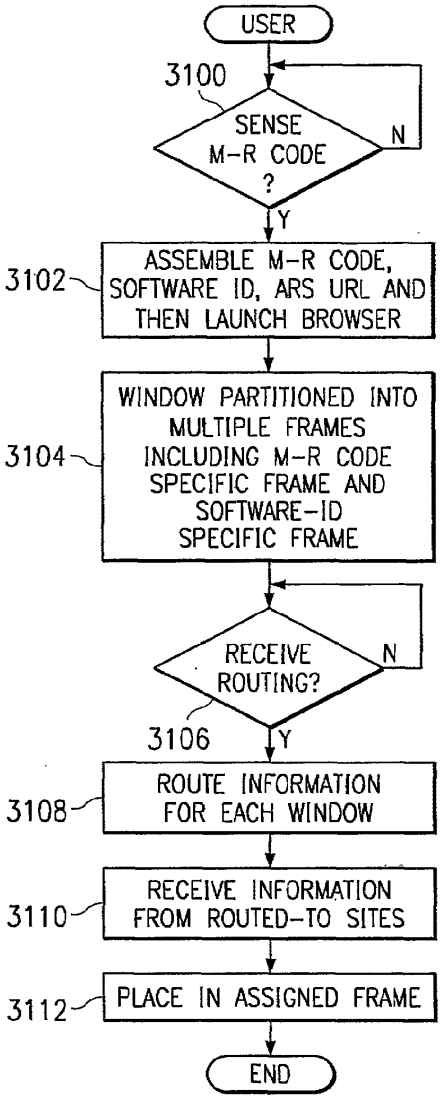
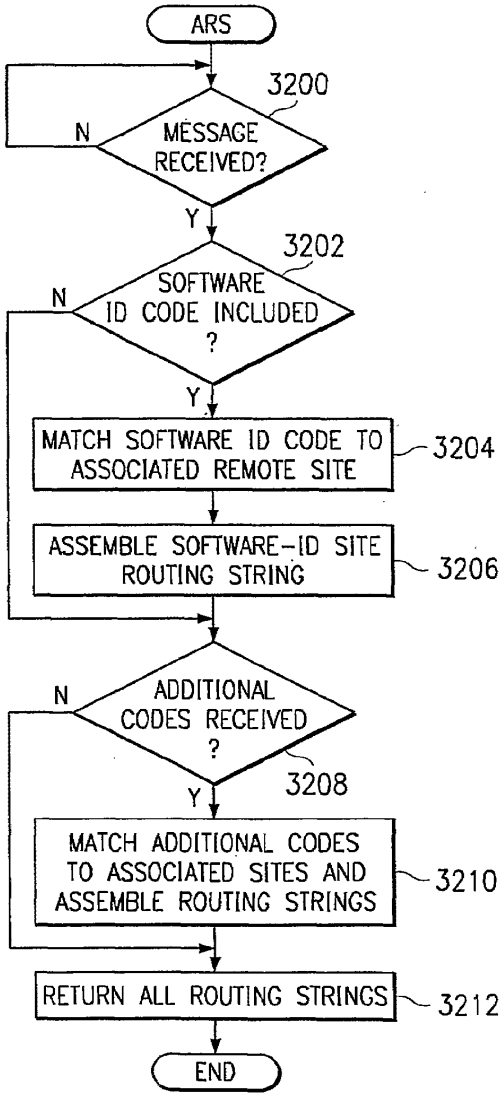

USER
3100
SENSE M-R CODE ?
N
Y
3102
ASSEMBLE M-R CODE, SOFTWARE ID, ARS URL AND THEN LAUNCH BROWSER
3104
WINDOW PARTITIONED INTO MULTIPLE FRAMES INCLUDING M-R CODE SPECIFIC FRAME AND SOFTWARE-ID SPECIFIC FRAME
RECEIVE ROUTING?
N
3106
Y
3108
ROUTE INFORMATION FOR EACH WINDOW
3110
RECEIVE INFORMATION FROM ROUTED-TO SITES
3112
PLACE IN ASSIGNED FRAME
END

FIG. 32


ARS
3200
N
MESSAGE RECEIVED?
Y
3202
N
SOFTWARE ID CODE INCLUDED ?
Y
MATCH SOFTWARE ID CODE TO ASSOCIATED REMOTE SITE
3204
ASSEMBLE SOFTWARE-ID SITE ROUTING STRING
3206
N
ADDITIONAL CODES RECEIVED ?
3208
Y
MATCH ADDITIONAL CODES TO ASSOCIATED SITES AND ASSEMBLE ROUTING STRINGS
3210
RETURN ALL ROUTING STRINGS
3212
END

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 October 2001 (11.10.2001)

PCT

(10) International Publication Number
WO 01/75629 A1

(51) International Patent Classification[7]: G06F 15/16, 15/173

(21) International Application Number: PCT/US01/10336

(22) International Filing Date: 29 March 2001 (29.03.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/193,836	31 March 2000 (31.03.2000)	US
60/193,755	31 March 2000 (31.03.2000)	US
60/193,737	31 March 2000 (31.03.2000)	US

(71) Applicant: NEOMEDIA TECHNOLOGIES, INC. [US/US]; 2201 Second Street, Suite 600, Fort Myers, FL 33901 (US).

(72) Inventors: HUNTER, Kevin, D.; 8381 Arborfield Court, Fort Myers, FL 33912 (US). DURST, Robert, T., Jr.; 6111 Tidewater Island Circle, Fort Myers, FL 33908 (US).

(74) Agent: BARKUME, Anthony, R.; Greenberg Traurig, LLP, 200 Park Avenue, New York, NY 10166 (US).

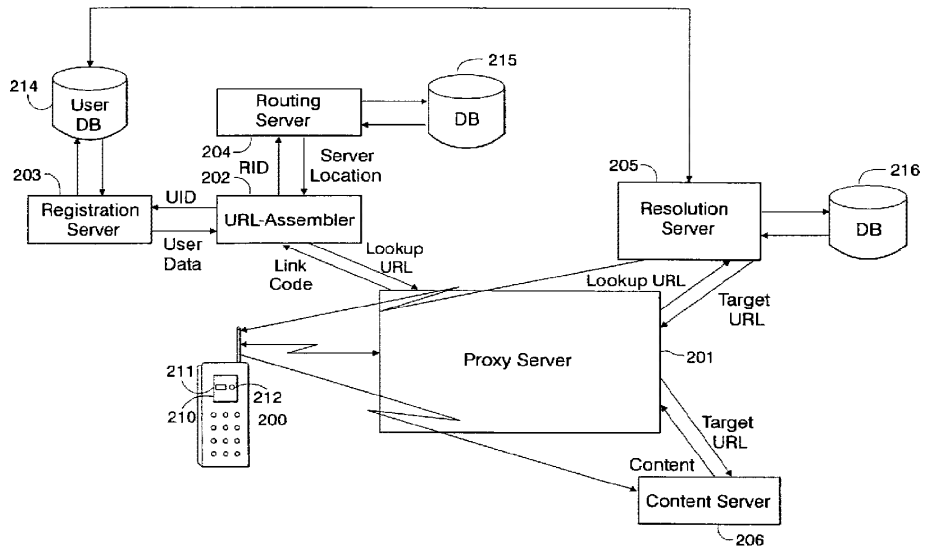
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: SYSTEM FOR ACCESSING INTERNET VIA WIRELESS DEVICE USING LINKAGE URL BAR-CODE



(57) Abstract: A system for accessing a primary content file (206) with a client device (200) that is browser-based (WAP); i.e., that does not require a plug-in type program to be executed on the client device in addition to the browser. URLs based on linkage codes entered into the client device (200) are assembled by a network-based URL-assembly server (202) rather than at the client device, wherein the client device is a wireless device supporting WML content or HTML content.

WO 01/75629 A1

SYSTEM FOR ACCESSING INTERNET VIA WIRELESS DEVICE USING LINKAGE URL BAR-CODE

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority from: (1) Hunter, "METHOD AND SYSTEM FOR SIMPLIFIED ACCESS TO INTERNET CONTENT ON A WIRELESS DEVICE", U.S. Provisional Application No. 60/193,737, filed March 31, 2000, the contents of which are incorporated herein by reference; (2) Hunter, et al., "SYSTEM FOR USING WIRELESS WEB DEVICES TO STORE WEB LINK CODES ON A LIST SERVER FOR SUBSEQUENT RETRIEVAL", U.S. Provisional Application No. 60/193,755, filed March 31, 2000, the contents of which are incorporated herein by reference; and (3) Hunter, "DEVICE-BASED ROUTING FOR WEB CONTENT RETRIEVAL", U.S. Provisional Application No. 60/193,836, filed March 31, 2000, the contents of which are incorporated herein by reference.

TECHNICAL FIELD

This invention relates to a method and system for simplified access of Internet content such as web pages through a wireless device such as a cellular telephone by entry of a linkage code that is associated with such Internet content.

BACKGROUND ART

Recently, a new generation of cell phones has been introduced that include provisions for Internet connectivity and "micro-browsers." Using this class of device, the cell phone user can directly access content on the World Wide Web, receive email, be notified of changes in "subscribed information" such as sports scores or stock prices, etc.

The constraints imposed on a "micro-browser" in a cell phone environment pose a unique problem for both the information provider as well as the user retrieving the

information. The development of the Wireless Application Protocol ("WAP") specification was specifically designed to address a number of fundamental differences between classic Internet and Web-based services and services on a wireless data network. These issues included the differences in needs and expectations as well as differences imposed by the device. That is, wireless devices will generally have less powerful CPU's, less memory and smaller displays than conventional computers. Wireless devices may have very different input devices. Wireless devices other than cell phones may be used that have very different capabilities. All of these issues have been addressed in the WAP specification and architecture. In particular, the WAP system is in no way restricted to cell phones - integration into other devices with wireless connectivity (e.g. Palm Pilots) was clearly anticipated. Thus, although this application will generally refer specifically to cell phones, anything described in that context would also apply to any other comparable wireless device, such as a Personal Digital Assistant ("PDA").

Figure 1 outlines the basic WAP architecture. Wireless devices are not directly "on the Internet" in the same sense as traditional computers. The fact that devices roam around, as well as the sheer number of devices expected to be deployed, discourage a solution in which each wireless device has its own IP address and communicates directly. In addition, the standard Internet protocols require a fair amount of overhead, which poses a problem on a channel with limited bandwidth. As a result, a new set of wireless protocols was developed. These include the Wireless Session Protocol ("WSP") and the Wireless Datagram Protocol ("WDP") which parallel the function of the TCP/IP and UDP Internet protocols. Wireless Telephone Application ("WTA") Servers "speak" these protocols natively, and can be directly accessed by wireless devices.

While for certain applications the requirement of a new class of server is acceptable, restricting wireless devices to this class of server would not adequately leverage the huge embedded base of Internet equipment. As such, the architecture includes WAP proxies, which serve as a bridge between the wireless network and the standard Internet. When a digital device attempts to access a resource via a standard URL, this request is passed to the WAP Proxy using wireless protocols. The WAP Proxy reformats this request into a standard HTTP 1.1 query, retrieves the content from the standard Web Server, and then passes the result back to the wireless device using the wireless protocol. Using this system, wireless devices can access any server on the Internet.

A web site that natively "understood" wireless devices would generally return content in the new Wireless Markup Language (WML), or possibly in the older Handheld Device Markup Language (HDML). Recognizing that achieving deployment of a second, parallel coding standard for documents might slow the penetration of the WAP technology, the WAP Architecture also includes provisions for filters that could translate standard HTML into WML automatically. These filters can be integrated into the WAP Proxy itself, or can exist between the web server and the WAP Proxy. This would, at least in theory, allow a wireless user to access any existing content on the web, even if the web site was not specifically designed for access by devices of this class.

A system and method for utilizing a link code or linkage code to cause a client computer to automatically access a web resource is disclosed in copending U.S. patent application serial number 09/543,178, filed on April 5, 2000 and incorporated by reference herein. In the system

described therein, the link code is a bar code that is scanned by a bar code scanner and input into a client software program that uses the decoded link code to request a URL template from an external server computer. The server takes the link code, returns the URL template, and the link client program assembles the URL using other data at the client. The URL is passed to a web browser, which then retrieves the web resource. This process may also be performed by manually entering a text string associated with the code, such as by entering a UPC number found at the bottom of a typical UPC barcode. This is a powerful way of utilizing a general purpose computer to automatically access a web resource without having to type in a lengthy URL.

It is noted that the system described above relies on the use of a client program running on the client computing device for obtaining, assembling, and then providing the URL to the web browser program. In general purpose personal computers, loading and running of such a client program is of course a typical and easily-done process. It would be desirable, however, to use this content retrieval technology with a client device such as a web-enabled cell phone, in which the use of add-on programs such as the linkage client is not easily accomplished. That is, with the proliferation of web-enabled cell phones and the like, it would be desirable to enable such devices to use such content retrieval methodologies without requiring adaptation of the software or firmware already present on such devices. One aspect of the invention described herein is thus a server-based URL assembly/retrieval methodology that requires no modification to existing web-enabled devices such as web-enabled cell phones or internet kiosks.

Another problem encountered by utilization of web-enabled cell phones for Internet access is that such devices have unique display capabilities. That is, one cannot

simply take any standard HTML web page and display it on a microbrowser or other such device. Thus, it is desired to be able to use the same linkage code in order to provide different types of content to different types of display devices. That is, it is desired to be able to format the content retrieved differently as a function of the device on which it will be displayed. As a result, a user entering a link code with a web-enabled cell phone will be automatically provided with WML or HDML content appropriate for display on that cell phone, while another user entering the same linkage code into a desktop computer will be provided with standard HTML content suitable for display on a large screen.

Although web-enabled cell phones can access web pages using the appropriate linkage code technology, one further problem is that much of the original content won't be displayable in the cell phone, since it will be in the form of big HTML pages. Alternatively, a user may be preoccupied and may simple wish to store a linkage code for subsequent retrieval. Therefore, it would be desirable to use the cell phone as a linkage code access device, without retrieving the associated content, but just for acquiring the linkage codes and uploading them to a code list server for later access by the user at a PC running the appropriate software.

DISCLOSURE OF THE INVENTION

In a first major aspect of the invention, provided is a method of and system for accessing a primary content file with a client device that is browser-based; i.e. that does not require a plug-in type program to be executed on the client device in addition to the browser. The user inputs into the client device a linkage code comprising a routing identification code and an item identification code, and the client device transmits to a URL-assembly server a data stream comprising the linkage code. The URL-assembly server

extracts the routing identification code from the data stream and then obtains a URL template associated with the routing identification code, the URL template comprising the name of a resolution server and at least one parameter field to be completed by the URL-assembly server. The URL-assembly server completes the URL template by filling in at least the item identification code and then sends the completed URL template to the resolution server named therein as a primary content URL request. The resolution server determines the location of the primary content file based on the item identification code and then provides the client device with the primary content file.

The URL-assembly server first obtains the URL template from a routing server. The URL-assembly server may then cache the URL template along with an expiration date for the URL template, which may be obtained from the routing server.

The data stream transmitted from the client device to the URL-assembly server may also further include a URL template selection code. In this event, then the URL-assembly server also extracts the URL template selection code from the data stream, and the URL template obtained by the URL-assembly server is also associated with the URL template selection code.

The URL template may be further completed by filling in least one parameter field with a device identification code, which may be included in the data stream transmitted from the client device to the URL-assembly server. The URL template may also be further completed by filling in least one parameter field with user data, which for example may be retrieved from a user database (populated by a user during a first registration process) located on a registration server.

The resolution server may directly return the primary content file, or it may redirect the client device to the primary content file by transmitting to the browser on the client device a primary URL for the primary content file. Alternatively, the primary URL may be sent to the client device via a proxy server, wherein the primary URL comprises an auto-request code that automatically redirects the client device to a content server containing the primary content file.

The linkage code input into the client device may for example be a bar code symbol that is scanned with a bar code scanning device coupled to the client device. The linkage code may also be a human-readable alphanumeric text string that may be typed in with a keypad connected to the client device.

The client device may be a wireless device such as a web-enabled cell phone, in which case a proxy server may used by the device for communication with the URL-assembly server, the resolution server and the content server.

Thus, in the method and system of the first major aspect of the invention, an existing browser-based client device such as a web-enabled cell phone or an internet kiosk, may be used to obtain content files from servers on the internet wherein the URL assembly steps are carried out on a server platform on the internet, instead of being done by a plug-in type module running with the client's browser.

In a second major aspect of the invention, provided is a method of and system for providing a primary content file to a client device wherein the primary content file will vary based on the display (or other) parameters of the client device, which is referred to as device-based routing. A linkage code comprising a routing identification code and

an item identification code is input into the client device, and the routing identification code and a client device identification code are then transmitted to a routing server. A URL template is obtained from the routing server, the URL template being associated with the routing identification code and the client device identification code, wherein the URL template includes the name of a resolution server and at least one parameter field to be completed by the client device. The URL template is completed by filling in the item identification code, such that the completed URL points to content suitable for display on the client device. The completed URL template is sent to the resolution server named therein to determine the location of the primary content file based on the item identification code and the client device identification code. A primary content URL that specifies the location of the primary content file is then sent to the client device, and the client device is redirected to a primary content server specified by the primary content URL.

For example, the client device may be a wireless device supporting WML content or HTML content, or it may be a personal computer supporting HTML content.

In a third major aspect of the invention, provided is a method of and system for providing a primary content file to a client device from a wireless device by using a stored code list service. In this case, the wireless device is used only to collect and transmit linkage codes to a list server (the linkage codes can be later accessed by any client device on the internet), and not to actually retrieve and display content associated with the linkage code. First, a linkage code is entered into a wireless transmitting device, wherein the linkage code comprises a routing identification code and an item identification code. The linkage code is then transmitted in a data stream to a list

server for storage therein, wherein the data stream includes a subscriber identification number associated with the wireless device. Later, a client device may be used to log into the list server, wherein the client device provides the subscriber identification number of the linkage code stored on the list server. The linkage code may then be retrieved from the list server and used to obtain a primary content file in accordance with one or more of various methods described herein.

The client device may be one of a group of devices comprising a personal computer, a web-based television, and a video console.

BRIEF DESCRIPTION OF THE DRAWING

FIG. 1 depicts a schematic overview of the wireless application protocol architecture.

FIG. 2 depicts a block diagram of an exemplary system of the present invention for a device directly connected to the Internet.

FIG. 2A depicts a block diagram of an alternative system of the present invention for a device connected to the Internet via the wireless application protocol.

FIG. 3A depicts a flow chart of how a linkage code is mapped to a content server.

FIG. 3B is a continuation of the flowchart of FIG. 3A.

FIG. 3C depicts a flow chart of how a new user is registered by the system of the invention.

FIG. 4 depicts an exemplary system of the present invention for the storage of linkage codes on list servers.

BEST MODE FOR CARRYING OUT THE INVENTION

The system of the present invention is a modification of the invention described in "SYSTEM AND METHOD OF USING MACHINE-READABLE OR HUMAN-READABLE LINKAGE CODES FOR ACCESSING NETWORKED DATA RESOURCES", copending U.S. Patent Application No. 09/543,178, filed April 5, 2000 by Hunter et al., previously incorporated herein by reference ("the copending application"). In the system described therein, a linkage code is a bar code that is scanned by a bar code scanner and input into a client software program that uses the decoded linkage code to request a URL template from an external server computer. The inputting of the code may also be performed by manually entering a text string associated with the code, such as by entering a UPC number found at the bottom of a typical UPC barcode. The linkage codes of the invention are not limited to UPC codes, however, and the invention supports European EAN codes, ISBN codes for books, as well as custom linkage code formats.

In a preferred embodiment of that invention, the linkage code includes two subcodes: a routing identification code ("RID") and an item identification code ("IID"). In the embodiment wherein the linkage code is a UPC code, the RID can be the manufacturer's portion of the UPC, whereas the IID can be the item code portion of the UPC. The client passes the RID to a routing server to obtain a URL link to a resolution server for that code, and the client completes the URL link by filing in the IID. The client then passes the completed URL link to the resolution server to obtain a target URL of content associated with the IID on the content server associated with the RID.

The two step resolution process allows for multiple resolution servers, thus providing scalability, with each server having its own database of target URLs. Since the

address of a resolution server for a particular RID changes infrequently with respect to number of times a user seeks to access the content server associated with the RID, the RID obtained from the routing server can be cached on the client for rapid lookup. The RID thus obtained can be associated with an expiration date so that the RID is periodically refreshed.

The system of the invention also includes a user database maintained by a registration server. The first time a user utilizes the invention to, for example, scan a UPC barcode to access a web site associated with the product, the user is directed to a registration procedure wherein the user is prompted to enter demographic data about him or herself. This data can include the user's name, address, age, gender, preferred language, and preferred interests. The registration server returns a user identification code ("UID") to the client, which caches it. The UID is passed to the routing server, which can then access the user data base and fill user data into the template URL. The template URL is returned to the client, which fills in the UID and IID to complete the lookup-URL. The client then passes the lookup-URL to the resolution server, which uses the user data along with a rules database to return a target-URL that addresses content specifically for that user. This feature of the invention is referred to as profiled routing.

Thus, the use of linkage codes is a powerful way of utilizing a general purpose computer to automatically access a web resource without having to type in a lengthy URL. Linkage codes are particularly useful in the context of wireless, hand-held web enabled devices such as cell phones, PDAs, or pocket personal computers ("pocket PCs"). Cell phones, for example, do not support the full alphabetic keyboard of a personal computer, and thus entering a full

URL for a web site is quite tedious. Most phones use a metaphor in which numeric buttons are pressed multiple times to scroll through several letters and/or punctuation marks, with either a button press or a pause indicating acceptance of the current letter. For example, www.amazon.com is entered on a cell phone numeric keypad as 99900262999966666002226666. On the other hand, the associated linkage code is merely 92801726. The all digit linkage code is shorter and easier to enter than the full URL, and much more intuitive to use. The advantage of linkage codes is even more apparent for those handheld devices that include barcode scanners, such as PDA's.

Although some wireless, hand-held web enabled devices, such as Palm Pilots or other PDAs, could easily be provided with the client plug-in required to map the linkage code into a URL, cell phones are not so easily adapted. There is also a large number of cell phones already in use. The inventors have thus found that it is preferable to locate this functionality on another server, referred to herein as a URL-assembly server. This enables any wireless device user to utilize linkage codes to access web content by merely accessing the appropriate page of the URL-assembly server that provides the mapping, without the necessity of installing the plug-in on the wireless client device.

Referring now to FIG. 2, an exemplary system configuration for a web-enabled device, such as a PDA that supports an HTML display, is depicted. Device 200 can execute a web browser whose interface is displayed in display area 210. When executing, the web browser can display the linkage code entry window, referred to as a go-window. The go-window includes a field 211 for entering a linkage code, and a button 212. A user can key in or write in a linkage code in field 211 and activate button 212 to find the associated web page. Alternatively, if device 200

supports a scanner 213, a barcode can be scanned in.

If device 200 is an Internet enabled device, such as a Palm VII PDA, it can transmit the linkage code just entered over the Internet to a URL-assembly server 202. The device 200 can also optionally transmit a user identification ("UID") to the URL-assembly server 202. If the device 200 is a WAP enabled cell phone that displays WML content, the transmission to the URL-assembly server 202 is typically mediated by a proxy server 201, shown in FIG 2A, that converts the WAP transmission into an HTTP compliant transmission. The URL assembly server 202 in turn communicates over the Internet with a registration server 203, which maintains a database of user information 214, and a routing server 204, which maintains a resolution server database 215. The URL-assembly server utilizes the RID portion of the linkage code, along with the UID, if available, to assemble a lookup URL in a manner described below. The lookup URL addresses a resolution server 205 that contains the target URL of the Internet content associated with the linkage code. The target URL received from the resolution server 205 redirects device 200 to the content server 206 containing the content associated with the linkage code.

The process by which linkage codes are mapped to content that is then downloaded is depicted in FIGS. 3A and 3B. A user begins the process by entering a linkage code in field 211 at step 300 in FIG 3A. The linkage code is transmitted to the URL-assembly server 202 at step 301. If a user is using the linkage code system for the first time, she would have to key in to the device 200 the name of the go-window in the traditional manner, by keying in the full name of the window, for example, www.paperclick.com. Once downloaded, however, the go-window can be bookmarked for easy subsequent retrieval.

The process by which a first-time user registers with the system is depicted in FIG. 3C. If the user is using a device that can transmit a unique device identifier, such as a PDA or cell phone using OpenWave's UP.Link proxy, she will be prompted at step 352 to register with the linkage code service. The user will be connected by the URL-assembly server 202 to the registration server 203. The registration server 203 can prompt the user at step 353 to enter various items of personal information, such as her name, address, age, gender, preferred language, and preferred interests. This information is stored at step 354 in user database 214. The user is assigned a UID so that she can be identified by the system. As part of this process, an entry can be made in the user database linking the UID to the unique device identifier. A given UID can even be linked multiple device identifiers.

Even if a client device does not support the transmission of a unique device identifier, it can still be identified to the system if the device's mini-browser supports standard authentication means, such as the storage of cookies. For this type of client, the registration server sends a cookie to the client's browser, which stores the cookie on the client device. Subsequent transmissions for the client to then URL-assembly server would then include the cookie.

The user data enables the linkage code system of the present invention to support the profiled routing feature of the client-based linkage system of the copending application. If, however, the device 200 does not support the transmission of a UID, the user of device 200 will remain anonymous to the linkage code system, and profiled routing is not available.

Continuing with FIG. 3A, once a linkage code has been received by the URL-assembly server 202, it is broken up at step 302 into its constituent parts, namely, the RID and the IID. The RID is passed to the routing server 204 at step 303 to obtain a URL template containing the address of the resolution server 205 associated with the IID. The resolution server 205 can actually be a front for any manufacturer's or vendor's server that can map a product code to a URL. This introduces the possibility that a given manufacturer or vendor could have a server for fielding WML queries that is different from servers fielding HTML queries. Since queries coming from the proxy server 201 typically indicate in the HTML header that the device 200 supports a WML browser, the URL-assembly server 202 can optionally include a URL template selection parameter in the data stream sent to the routing server so that a WML oriented template is returned by the routing server 204 to the URL-assembly server 202. The template selection parameter can also be used to ensure that the WML content ultimately returned to the user is not encapsulated in a frameset, as framesets are not supported by WML.

If the device 200 has been previously registered with the system, its UID will be included in the transmission to the URL-assembly server 202. In this situation, the URL-assembly server at step 304 passes the UID to the registration server which in turn uses the UID to retrieve user data for that user from the user database 214, and returns that data to the URL-assembly server 202.

The URL-assembly server 202 completes the URL-template at step 305. The URL template returned by the routing server 204 has at least one field for the URL-assembly server 202 to fill in. A typical URL template will look something like:

http://resolve.paperclick.com:8080/all/cmd?CMD=GET&TYPE=^TYP

E^&RID=^RID^&IID=^IID^&CODE=^CODE^, wherein the fields delimited by carets ("^") are to be filled in by the URL-assembly server. In the example shown, the fields to be filled in are the code type, the RID, the IID, and the full linkage code. If the linkage code is a UPC code equal to 051111128817, the RID would be 051111, the IID would be 12881, and the completed URL would be http://resolve.paperclick.com:8080/all/cmd?CMD=GET&TYPE=UPC&RID=051111&IID=12881&CODE=051111128817. There can also be fields for the UID, user data retrieved from the user database maintained by the registration server, and the template selection parameter. This list of fill-in codes is illustrative, and more fill-in fields can be supported and be within the scope of the invention.

The URL template also includes a field for a parameter indicative of the display device, i.e. what markup language the display device supports. This parameter could be the template selection parameter, or it could be a separate parameter. This enables the resolution server to list the addresses of multiple versions of a given page, a feature referred to a device-based routing. Thus, publishers can host web content on multiple formats accessible with different URLs, but use the same linkage code to access the content. Users are dynamically routed to the proper content based on the characteristics of the retrieving device.

The completed URL, referred to as a lookup-URL, is a reference to both a particular resolution server and an entry in that resolution server. The resolution server can be a front for any server, such as a vendor's server, that can map the IID to appropriate content on the Internet. The lookup-URL is returned at step 306 to the device 200, or the proxy server 201 if device 200 is a WAP device, and redirects the device 200, or the proxy server 201, to the resolution server. Continuing onto FIG. 3B, the resolution

server 205, at step 307, finds an appropriate target URL based on the information contained in the lookup-URL: the RID, the IID, and the user data if the user is registered. This ensures that the content customized to the user is subsequently returned. The resolution server includes lookup-tables and rules that ensure that a target URL to a content server 206 containing a web page in the correct display language is returned to the sender. The use of rules and tables to map the lookup-URL to a target-URL for content appropriate for a particular user is described in the copending application, and need not be repeated herein. The skilled artisan can easily extend the rules disclosed therein to support the device based routing feature of the present invention.

The target-URL returned by the resolution server normally redirects the sending device, either device 200 or proxy server 201, to the content server 206. In the case of a WAP compliant cell phone, however, having the proxy server perform the redirection means that the cell phone's mini-browser will not know of the redirection. Thus, when the cell phone device 200 receives the content, it would think the content had come from the server specified by the lookup URL, i.e. the resolution server 205, not the content server 206 specified by the target URL. If the returned content includes a relative URL or image reference, the device 200 will issue a request to the resolution server 205, not the content server 206. Therefore, the redirection to the content server 206 is not performed by the proxy server 201. Instead, at step 311, a data stream is returned to the WAP device 200 that includes the target-URL hyperlink along with an auto-click code to force the device 200 at step 312 to automatically make the request to the content server 206. If the device is directly connected to the Internet, that device is redirected at step 309 to the content server 206. Finally, the content is downloaded to the device 200 at step

310.

In addition to the functionality described above, the system of the present invention supports the demographic reporting, obfuscation/de-obfuscation of the UID, and profiled routing features disclosed in the copending application. In addition, the device based routing feature of the present invention can also be included with the invention of the copending application. The linkage client software disclosed in the copending application can easily be modified to include a device indicator field in the data stream transmitted to the routing server. This enables the routing server to select a URL template appropriate for the display device.

One application of profiled routing is the ability to streamline registration for cell-based services. Where there are user-specific parameters such as presentation language, a user registering for a service via a linkage code could have profile information passed from the user database 214 into the service registration process. This would potentially allow the registration form to be pre-populated with the user's information, thus allowing the user to simply confirm the information, rather than having to laboriously enter it.

In many cases, a cell phone user, because of the sparse nature of the WML or HDML display as compared with an HTML form, or because the user is preoccupied with another task, may not want to actually download content directly to her cell phone upon entering a linkage code, but may wish to save the linkage code for subsequent retrieval by a personal computer that supports HTML. In another aspect of the invention, a registered user can store linkage codes on a list server for subsequent retrieval. Referring now to Figure 4, by using a simple WML or HDML form 401 on the cell

phone 400, a linkage code is entered into field 408 in the form 401 via the keypad 403, and then submitted via a store button 409 over the wireless network 404 and Internet 405 to a list server 406, where it would be stored. The wireless transactions pass on the user's UID, so the system can use this to indicate which list the code should be added to. Later, when the user sits down at his or her PC 407, he or she can log into the server 406, and download the accumulated stored codes to the PC 407, just as if they had used the linkage client software. A user does not even need a PC - he or she could retrieve the codes via a WebTV, video game console (the newest generation of video game consoles apparently have browsers and modems built in) or any other device that has an embedded browser. The user's login/password can be used as the authentication means, and can be tied to their UID via the registration process.

By using the stored linkage code list service described herein, a user can download stored linkage codes lists to a client device anywhere in the world. The user can even upload linkage codes from a desktop client to the list server, then download them onto any other client device, such as a laptop computer via the web.

The system of the present invention is currently implemented on a Windows NT platform, although the system can be adapted to operate on other operating systems, such as Unix or Linux, or the Macintosh. The registration, routing and resolution servers are currently implemented as stand-alone programs, written in C++, that run as services under Windows NT, and the URL-assembly server is currently implemented as a component of the routing server. Other implementation are, of course possible, and the servers could be implemented as ISAPI DLLs running on a web server that communicate with a Microsoft SQL database server via ODBC, as CGI programs or as Java servlets. The routing

server can also be implemented as a stand-alone program. Although these components have been described as if they are physically distinct machines, the skilled artisan will understand that they can be distinct processes running on the same machine.

The system of the invention is not limited to the embodiment disclosed herein. It will be immediately apparent to those skilled in the art that variations and modifications to the disclosed embodiment are possible without departing from the spirit and scope of the present invention. The invention is defined by the appended claims.

WHAT IS CLAIMED IS:

1. A method of accessing a primary content file with a client device comprising the steps of:

(a) inputting into the client device a linkage code comprising a routing identification code and an item identification code;

(b) transmitting from the client device to a URL-assembly server a data stream comprising the linkage code;

(c) extracting by the URL-assembly server the routing identification code from the data stream;

(d) obtaining by the URL-assembly server a URL template associated with the routing identification code, the URL template comprising the name of a resolution server and at least one parameter field to be completed by the URL-assembly server;

(e) completing at the URL-assembly server the URL template by filling in the at least one parameter field;

(f) sending the completed URL template to the resolution server named therein as a primary content URL request;

(g) determining at the resolution server the location of the primary content file based on the item identification code; and

(h) the resolution server providing the client device with the primary content file.

2. The method of claim 1, wherein the URL template is obtained from a routing server.

3. The method of claim 2, further comprising the step of caching the URL template on the URL-assembly server, along with an expiration date for the URL template.

4. The method of claim 3, wherein the expiration date for the URL template is obtained from the routing server.

5. The method of claim 3, further comprising the step of retrieving the URL template from the routing server when the current date is later than the expiration date.

6. The method of claim 1, wherein the data stream transmitted from the client device to the URL-assembly server further comprises a URL template selection code.

7. The method of claim 6, further comprising the step of the URL-assembly server extracting the URL template selection' code from the data stream.

8. The method of claim 7, wherein the URL template obtained by the URL-assembly server is also associated with the URL template selection code.

9. The method of claim 1, wherein the at least one parameter field is filled in with the item identification code by the URL-assembly server

10. The method of claim 1, wherein the URL template is further completed by filling in the least one parameter field with a device identification code.

11. The method of claim 10, wherein the device identification code is included in the data stream transmitted from the client device to the URL-assembly server.

12. The method of claim 1, wherein the URL template is further completed by filling in least one parameter field with user data.

13. The method of claim 12, wherein the user data is retrieved from a user database located on a registration server.

14. The method of claim 13, wherein the user database is populated by a user during a first registration process.

15. The method of claim 13, further comprising a second registration process wherein the user uses the linkage code to register for a service, and the second registration process uses user data retrieved from said user database.

16. The method of claim 1, wherein the resolution server provides the client device with the primary content file by transmitting to the client device a primary URL for the primary content file, the primary URL comprising an auto-request code that automatically redirects the client device to a content server containing the primary content file.

17. The method of claim 16, wherein the primary URL is sent to the client device via a browser.

18. The method of claim 16, wherein the primary URL is sent to the client device via a proxy server.

19. The method of claim 1, wherein the linkage code is a bar code symbol, and wherein the step of inputting comprises the step of scanning the bar code symbol with a bar code scanning device connected to the client device.

20. The method of claim 1, wherein the linkage code is a human-readable alphanumeric text string, and wherein the step of inputting comprises the step of typing in the alphanumeric text string with a keypad connected to the client device.

21. The method of claim 1, wherein the client device is a wireless device.

22. The method of claim 21, further comprising a proxy server, by means of which the wireless device communicates with the URL-assembly server, the resolution server and the content server.

23. A computer system for accessing a primary content file on a primary content server over a computer network with a client device, comprising:

(a) a client device interconnected to the computer network;

(b) a URL-assembly server interconnected to the computer network;

(c) a resolution server interconnected to the computer network; wherein

the client device comprises:

means for inputting a linkage code comprising a routing identification code and an item identification code;

means for transmitting a data stream comprising the linkage code to the URL-assembly server;

the URL-assembly server comprises:

means for extracting the routing identification code from the data stream received from the client device;

means for obtaining a URL template associated with the routing identification code, the URL template comprising the name of a resolution server and at least one parameter field to be completed by the URL-assembly server;

means for completing the URL template by filling in the at least one parameter field;

means for sending the completed URL template to the resolution server named therein as a primary content URL request; and

the resolution server comprises:

means for determining the location of the primary

content file based on the item identification code; and

means for providing the client device with the primary content file.

24. The computer system of claim 23, further comprising a routing server from which the URL-assembly server obtains the URL template.

25. The computer system of claim 24, wherein the URL-assembly server caches the URL template, along with an expiration date for the URL template.

26. The computer system of claim 25, wherein the expiration date for the URL template is obtained from the routing server.

27. The computer system of claim 25, wherein the URL-assembly server further comprises means for retrieving the URL template from the routing server when the current date is later than the expiration date.

28. The computer system of claim 23, wherein the data stream transmitted from the client device to the URL-assembly server further comprises a URL template selection code.

29. The computer system of claim 28, wherein the URL-assembly server further comprises means for extracting the URL template selection code from the data stream.

30. The computer system of claim 29, wherein the URL template obtained by the URL-assembly server is also associated with the URL template selection code.

31. The computer system of claim 23, wherein the URL-assembly server further comprises means for completing the

URL template by filling in least one parameter field with the item identification code.

32. The computer system of claim 23, wherein the URL-assembly server further comprises means for completing the URL template by filling in least one parameter field with a device identification code.

33. The computer system of claim 32, wherein the client device further comprises means for including the device identification code in the data stream transmitted from the client device to the URL-assembly server.

34. The computer system of claim 23, wherein the URL-assembly server further comprises means for further completing the URL template by filling in least one parameter field with user data.

35. The computer system of claim 34, further comprising a registration server comprising a user database, wherein the user data is retrieved from the user database.

36. The computer system of claim 35, wherein the user database is populated by a user during a first registration process.

37. The computer system of claim 35, further comprising a second registration process wherein the user uses the linkage code to register for a service, and the second registration process uses user data retrieved from said user database.

38. The computer system of claim 23, wherein the resolution server comprises means for providing the client device with the primary content file by transmitting to the client device a primary URL for the primary content file,

the primary URL comprising an auto-request code that automatically redirects the client device to a content server containing the primary content file.

39. The computer system of claim 38, wherein the primary URL is sent to the client device via a browser.

40. The computer system of claim 38, wherein the primary URL is sent to the client device via a proxy server.

41. The computer system of claim 23, wherein the means for inputting a linkage code comprises a bar code scanning device for scanning a linkage code in the form of a bar code symbol.

42. The computer system of claim 23, wherein the means for inputting a linkage code comprises a keypad for entering a linkage code in the form of a human-readable alphanumeric text string.

43. The computer system of claim 23, wherein the client device is a wireless device.

44. The computer system of claim 43, further comprising a proxy server, by means of which the wireless device communicates with the URL-assembly server, the resolution server and the content server.

45. A method of providing a primary content file to a client device comprising the steps of:

(a) inputting into the client device a linkage code comprising a routing identification code and an item identification code;

(b) transmitting to a routing server the routing identification code and a client device identification code, and obtaining from the routing server a URL template

associated with the routing identification code and the client device identification code, the URL template comprising the name of a resolution server and at least one parameter field to be completed by the client device;

(c) completing the URL template by filling in the item identification code, the completed URL pointing to content suitable for display on the client device;

(d) sending the completed URL template to the resolution server named therein to determine the location of the primary content file based on the item identification code and the client device identification code; and

(e) sending a primary content URL that specifies the location of the primary content file to the client device and redirecting the client device to a primary content server specified by the primary content URL.

46. The method of claim 45, further comprising the step of providing the primary content file to the client device from the primary content server.

47. The method of claim 45, wherein the client device is a wireless device supporting WML content.

48. The method of claim 45, wherein the client device is a wireless device supporting HTML content.

49. The method of claim 45, wherein the client device is a wireless device supporting HDML content.

50. The method of claim 45, wherein the client device is a personal computer supporting HTML content.

51. A system for providing a primary content file to a client device over a computer network, comprising:

(a) a client device interconnected to the computer network;

(b) means for inputting into the client device a linkage code comprising a routing identification code and an item identification code;

(c) means for transmitting to a routing server the routing identification code and a client device identification code, and means for obtaining from the routing server a URL template associated with the routing identification code and the client device identification code, the URL template comprising the name of a resolution server and at least one parameter field to be completed by the client device;

(d) means for completing the URL template by filling in the item identification code, the completed URL pointing to content suitable for display on said client device;

(e) means for sending the completed URL template to the resolution server named therein to determine the location of the primary content file based on the item identification code and the client device identification code; and

(f) means for sending a primary content URL that specifies the location of the primary content file to the client device and redirecting the client device to a primary content server specified by the primary content URL.

52. The system of claim 51, further comprising means for providing the primary content file to the client device from the primary content server.

53. The system of claim 51, wherein the client device is a wireless device supporting WML content.

54. The system of claim 51, wherein the client device is a wireless device supporting HDML content.

55. The system of claim 51, wherein the client device is a wireless device supporting HTML content.

56. The system of claim 51, wherein the client device is a personal computer supporting HTML content.

57. A method of providing a primary content file to a client device from a wireless device, said method comprising the steps of:

(a) entering a linkage code into a wireless transmitting device, wherein the linkage code comprises a routing identification code and an item identification code;

(b) transmitting the linkage code in a data stream to a list server for storage therein, wherein said data stream includes a subscriber identification number associated with the wireless device;

(c) logging in to the list server from a client device, wherein the client device provides the subscriber identification number of the linkage code stored on the list server; and

(d) retrieving the linkage code from the list server.

58. The method of claim 57, wherein a user provides the client device with the subscriber identification number.

59. The method of claim 57, wherein the client device is one of a group of devices comprising a personal computer, a web-based television, and a video console.

60. The method of claim 57, further comprising the step of utilizing the linkage code to enable the client device to retrieve a primary content file associated with the item identification code from a content server associated with the routing identification code.

61. A system for providing a primary content file to a client device from a wireless device over a computer network, comprising: