# Multimedia Data-Embedding and Watermarking Technologies

MITCHELL D. SWANSON, MEMBER, IEEE, MEI KOBAYASHI, AND
AHMED H. TEWFIK, FELLOW, IEEE

*Invited Paper*

*In this paper, we review recent developments in transparent data embedding and watermarking for audio, image, and video. Data-embedding and watermarking algorithms embed text, binary streams, audio, image, or video in a host audio, image, or video signal. The embedded data are perceptually inaudible or invisible to maintain the quality of the source data. The embedded data can add features to the host multimedia signal, e.g., multilingual soundtracks in a movie, or provide copyright protection. We discuss the reliability of data-embedding procedures and their ability to deliver new services such as viewing a movie in a given rated version from a single multicast stream. We also discuss the issues and problems associated with copy and copyright protections and assess the viability of current watermarking algorithms as a means for protecting copyrighted data.*

***Keywords*—***Copyright protection, data embedding, steganography, watermarking.*

## I. INTRODUCTION

The past few years have seen an explosion in the use of digital media. Industry is making significant investments to deliver digital audio, image, and video information to consumers and customers. A new infrastructure of digital audio, image, and video recorders and players, on-line services, and electronic commerce is rapidly being deployed. At the same time, major corporations are converting their audio, image, and video archives to an electronic form.

Digital media offer several distinct advantages over analog media: the quality of digital audio, image, and video signals is higher than that of their analog counterparts. Editing is easy because one can access the exact discrete locations that should be changed. Copying is simple with no loss of fidelity. A copy of a digital media is identical to the original. Digital audio, image, and videos are easily transmitted over networked information systems.

These advantages have opened up many new possibilities. In particular, it is possible to hide data (information) within digital audio, image, and video files. The information is hidden in the sense that it is perceptually and statistically undetectable. With many schemes, the hidden information can still be recovered if the host signal is compressed, edited, or converted from digital to analog format and back.

As we shall see in Section II, pure analog data-hiding techniques had been developed in the past. However, these techniques are not as robust as most of the digital data hiding techniques that we review in this paper. Furthermore, they cannot embed as much data in a host signal as the digital approaches.

Digital data embedding has many applications. Foremost is passive and active copyright protection. Many of the inherent advantages of digital signals increase problems associated with copyright enforcement. For this reason, creators and distributors of digital data are hesitant to provide access to their intellectual property. Digital watermarking has been proposed as a means to identify the owner or distributor of digital data.

Data embedding also provides a mechanism for embedding important control, descriptive, or reference information in a given signal. This information can be used for tracking the use of a particular clip, e.g., for pay-per-use applications, including billing for commercials and video and audio broadcast, as well as Internet electronic commerce of digital media. It can be used to track audio or visual object creation, manipulation, and modification history within a given signal without the overhead associated with creating

Manuscript received July 15, 1997; revised January 15, 1998. The Guest Editor coordinating the review of this paper and approving it for publication was A. M. Tekalp. This work was supported in part by the Air Force Office of Scientific Research under Grant AF/F49620-94-1-0461 and in part by the Advanced Research Project Agency under Grant AF/F49620-93-1-0558.

M. D. Swanson is with Cognicity, Inc., Minneapolis, MN 55344 USA (e-mail: swanson@cognicity.com).

M. Kobayashi is with the Graduate School of Mathematical Sciences, University of Tokyo and IBM Tokyo Research Laboratory, Yamato-shi, Kanagawa-ken 242 Japan (e-mail: mei@trl.ibm.co.jp).

A. H. Tewfik is with Cognicity, Inc., Minneapolis, MN 55344 USA (e-mail: tewfik@cognicity.com) and the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: tewfik@ece.umn.edu).

Publisher Item Identifier S 0018-9219(98)03519-1.

a separate header or history file. It can also be used to track access to a given signal. This information is important in rights-management applications.

Data embedding is also ideally suited for covert communications. Data embedding can securely hide large amounts of potentially encrypted information in audio, image, and video data.

A most interesting application of data embedding is providing different access levels to the data. For example, the amount of detail that can be seen in a given image can be controlled. A person with a high access level can see details that another person with a lower access level would not see. Similarly, data embedding allows users to tailor a video to their needs, e.g., by watching a movie broadcast over a single channel in a particular rating or in a given language. In this case, data embedding is used to embed extra scenes and multilingual tracks in a given version of the movie that is broadcast [84]. In a sense, data embedding then provides some of the capability of digital video disc (DVD) in a broadcast environment with no extra bandwidth or storage requirements.

Most data-embedding algorithms can extract the hidden data from the host signal with no reference to the original signal. In some scenarios, an original is available to the detection algorithm. Typically, data-embedding algorithms that use the original signal during detection are robust to a larger assortment of distortions. The detection algorithm may “subtract off” the original signal from the received signal prior to data detection. Registration may also be used by receivers to compare the received signal with the original to correct scaling, rotation, and other distortions prior to data detection. Some data-embedding algorithms require access to the original data to derive parameters, e.g., hash values, that are required during detection. As different data-embedding applications have different requirements, we distinguish between these cases in this review.

Note also that most data-embedding algorithms assume that it is desirable to have secure data-embedding and extraction procedures. Specifically, a secret key typically determines how the data are embedded in the host signal. A user needs to know that key to extract the data. Knowledge of that key and the embedding algorithm would also allow the user to overwrite or erase the embedded information. In some applications, e.g., copy control in DVD or fraud detection by a recipient of the signal, it is desirable to give all users access to the embedded data *without* enabling them to change or remove that data. This problem has been addressed in cryptography. However, the solutions developed in cryptography cannot be applied directly in the watermarking or data-hiding context. In fact, to date, no satisfactory solution to that problem has been proposed within the data-embedding or watermarking literature. Some pioneering work in that area is described in [33].

The goal of this paper is to present an overview of the challenges and issues that need to be addressed by successful watermarking and data-embedding techniques and the current state of the art. Data-embedding and watermarking research builds on ideas and concepts developed in cryptography, communications theory, algorithm design, and signal processing. The data-embedding problem is inherently more difficult than any of the problems that have traditionally been addressed in these fields. All data-embedding algorithms combine and extend in a sense many of the solutions developed in these areas. Most of the published work on data embedding that has appeared in technical journals and conferences focuses on image and video data. On the other hand, most of the published work on audio data embedding has appeared in the patent literature. The coverage of this review in the audio, image, and video areas is basically proportional to the existing journal and conference literature in these three fields.

In the next section, a brief historical overview of the field is given. In particular, we relate some of the techniques that have been proposed recently in the areas of data embedding and watermarking to older steganographical techniques. In Section III, the basic requirements of data embedding and watermarking are addressed. We discuss the different security and robustness requirements of data-embedding applications. We also review the deadlock problem that arises in ownership identification and describe two solutions. Data embedding and watermarking in digital media are possible because of the limitations of the human auditory and visual systems. We review some properties of human auditory and visual perception in Section IV. Following this review, we describe the principles that underlie current data-embedding approaches. We provide examples to illustrate the capability of today’s technology. Sections V–VII present image, audio, and video data-embedding techniques, respectively. We conclude the paper with a brief overview of visible watermarking approaches.

## II. History

Data-embedding and watermarking techniques are particular embodiments of steganography (from the Greek words $\sigma\tau\epsilon\gamma\alpha\nu\omega$ or *stegano* for “covered” and *graphos*, “to write”). In contrast to cryptography, which focuses on rendering messages unintelligible to any unauthorized persons who might intercept them, the heart of steganography lies in devising astute and undetectable methods of concealing the messages themselves.

Marking of documents may have evolved alongside human writing during the dawn of Western civilization. Since knowledge of writing was often restricted to a privileged and powerful class, the need to conceal messages from traitors and enemies within these circles appears to have been a serious concern. In a historical text on coding [43], Kahn traces the roots of secret writing back 4000 years to the banks of the Nile, where hieroglyphic symbol substitutions were used to inscribe information in the tomb of a nobleman, Khnumhotep II. The intent of the substitutions is ambiguous. The earliest allusion to secret writing in the West with concrete evidence of intent appears in Homer’s *Iliad* [35]. Steganographic methods per se made their recorded debut a few centuries later in several tales by Herodotus [34], although the term *steganography* does not come into use until many centuries later, in the 1500's, after the appearance of Trithemius' book on the subject, *Steganographia*. Ancient references to secret writing and steganography also appear in Asia. Indian literature is replete with references as well as explicit formulas for secret writing. Kautilya's *Artha-śāstra* which dates back to 321–300 B.C., the *Lalita-Vistara*, and Vātsāyana's *Kāma-sūtra* are a few of the more famous examples. In fact, the study of many different types or cryptography, not just steganography, flourished in ancient India. In ancient China, military and diplomatic rulers wrote important messages on thin sheets of silk or paper. For secure transport, the sheets were rolled into balls, covered with wax, and swallowed by or placed in the rectum of messengers. Less sensitive, routine messages were usually memorized, then transmitted orally by a messenger.

It is interesting to note that many of the steganographical techniques that had been devised in the past have recently reappeared in the data-embedding and watermarking literature. For example, a class of steganographic techniques relies on using semagrams (*sema* for "sign" and *gramma* for "something written or drawn"), i.e., very slight physical differences in appearance such as special fonts, punctuation marks, or very fine dots. A well-known semagram approach consists of marking text using barely visible pin pricks, small dots, and dashes. The technique was suggested by Aenas the Tactician and used during the Renaissance and up through the twentieth century. (A modern adaptation of the technique is used in WitnesSoft by Aliroo,[1] which marks electronic documents during printout with barely visible dots, which can only be picked up by high-resolution scanners.) Embedding of messages by lowering specified letters and varying spacing between words appears time and again throughout history. Recently, the technique has been revisited in a digital context by scientists who are investigating digital watermarking of text files [6], [49], [56], [93], [98]. Examples of nontextual semagrams are equally replete. Spies have embedded messages in Morse code in drawings, e.g., landscapes with short and tall leaves of grass representing dots and dashes. Graphs have been disguised in mazes in a puzzle book, as have images in autostereograms. A modern extension of these techniques is the embedding of marks, such as "VOID," in an image or document that appear only when photocopied [58]. An early example of copyright or authorship information in musical scores was practiced by Bach. Bach embedded his name in many of his pieces (e.g., his organ chorale *Vor deinem Thron*) using null cipher coding by spelling out *B-A-C-H* in notes (where B-flat represents *B*, and B represents *H*) or by counting the number of occurrences of a note (one occurrence for *A*, two for *B*, three for *C*, and eight for *H*).

## III. Requirements

As mentioned in the introduction, data embedding can be used in many different applications. Obviously, different applications will have different requirements. Therefore, there is no unique set of requirements that all data-embedding techniques must satisfy. Nevertheless, certain requirements must be satisfied in several application areas. In this section, we shall review some of these requirements and indicate when they are important.

### *A. Perceptual Transparency*

The focus of this paper is on perceptually undetectable or transparent data-embedding and watermarking techniques. In many applications, such as copyright and usage tracking, embedding metadata or additional information, the algorithms must embed data without affecting the perceptual quality of the underlying host signal. In some applications, such as low-quality browsing of signals prior to purchasing, perceptually detectable watermarks have been used. We shall have more to say about such watermarks in Section VIII.

A data-embedding procedure is truly imperceptible if humans cannot differentiate between the original host signal and a host signal with inserted data. Typically, blind tests are used to assess the perceptual transparency of data-embedding procedures. In such tests, subjects are presented randomly with signals with and without embedded data and asked to determine which signal has a perceptually higher quality. A probability of selecting the signal with no embedded data that is roughly equal to 50% is indicative of perceptual transparency. Note that the blind tests must assess also the effect of several of the typical modifications that the signal may undergo. For example, digital pictures typically undergo a sharpening or high-pass filtering operations. Data embedding should not produce artifacts that are perceptually dissimilar from those that may be seen in an untampered image.

### *B. Recovery of Data with or Without Access to Original Signal*

In some applications, such as copy tracking and copyright protection, the data-extraction algorithms may use the original signal to decode the embedded data. However, in most applications, data-embedding algorithms do not have access to the original audio, image, or video signal while extracting the embedded signal. This inability to access the original signal limits the amount of data that can be embedded in a given host signal. It also renders data extraction more difficult.

Specifically, the embedded data may be considered as information transmitted on a communication channel and corrupted by strong interference and channel effects. The strong interference consists of the host signal. Channel effects correspond to postprocessing operations. Most data-extraction procedures are inherently projection techniques on a given direction. Ideally, a larger projection value will indicate the presence of one type of data, e.g., a binary symbol or a watermark that represents an author. A segment of the host signal that is highly correlated with the projection direction will provide a false detection. Fur-

[1] See http://www.aliroo.com.

thermore, it may be impossible to modify that segment to reduce its correlation with the projection direction without affecting the perceptual quality of the host signal. Hence, the algorithm may be unable to embed useful data into that segment.

Note that the projection direction cannot be easily changed since the decoder does not have access to the original host signal. Any change in that direction must be accomplished through an algorithm that uses the received modified host signal. Note also that the probability of getting a high correlation between an arbitrary segment of the host signal and the projection direction decreases as the size of the segment increases. However, as that size increases, the amount of data that can be embedded in the host signal decreases.

Postprocessing effects can complicate the detection process. For example, synchronization problems may arise as a consequence of temporal and spatial rescaling, cropping, resampling, rotation, etc. Many modifications lead to new signals, which have a different number of samples than the original signal with embedded data. To extract the embedded information, the extraction algorithm must adapt to the new signal with fewer samples automatically or access the original to register the signal. Note, however, that loss of synchronization does not imply that the embedded data have been erased. If complexity is not an issue, the data can still be recovered.

### C. *Bit Rate of Data-Embedding Algorithm*

Some applications of data embedding, e.g., insertion of a serial number or author identification or fraud detection, require that relatively small amounts of information be incorporated repeatedly in the signal. On the other hand, many envisioned applications of data embedding, e.g., embedding a smaller image into a larger image or embedding multiple speech signals into a video, require a lot of bandwidth. In these cases, the algorithms must be able to embed an amount of data that is a significant fraction of the amount of data in the host signal. As mentioned above, the ability to embed large quantities of data in a host signal will depend critically on how the embedding algorithm can adapt its insertion strategy to the underlying host signal.

### D. *Robustness*

Some data-embedding applications may take place in an error-free or lossless environment. For example, the embedded data may provide digital object identifiers for use in clean signals residing in a controlled data base. In these situations, robustness to signal degradations is not important. In many cases, however, lossy signal-processing operations may be present in the system. For example, in most applications involving storage or transmission of an image, a lossy coding operation is performed on the image to reduce bit rates and increase efficiency. Digital data are readily modified and manipulated using computers and widely available software packages, e.g., Adobe Photoshop or Premiere. Operations that damage the host signal also damage the embedded data. Furthermore, third parties may attempt to modify the host signal to thwart detection of the embedded data.

Designers of robust data-embedding procedures have focused on several types of malicious or incidental host signal modifications. These modifications include:

- additive Gaussian or non-Gaussian noise;
- linear filtering, e.g., low-pass and high-pass filtering;
- nonlinear filtering, e.g., median filtering;
- compression, e.g., Joint Photographic Experts Group (JPEG), Moving Picture Experts Group (MPEG), wavelet;
- local exchange of samples, e.g., permutations;
- quantization of sample values;
- rotation;
- spatial or temporal scaling;
- removal or insertion of samples, pixels, or video frames;
- temporal averaging, e.g., averaging of successive video frames;
- swapping of data, e.g., swapping of successive video frames;
- averaging multiple watermarked copies of a signal;
- digital–analog (D/A) and analog–digital (A/D) conversions, e.g., printing and scanning or tape recording and redigitization.

Note that software to test robustness and remove data embedded from images is widely available on the Internet. In particular, the UnZign[2] and StirMark [45] programs have shown remarkable success in removing data embedded by commercially available programs. The algorithms generally apply minor geometric distortions, e.g., slight stretching, shearing, shifting, and/or rotations, to the image and then resample the image using bilinear interpolation. The resulting image looks perceptually similar to the original signal with embedded data.

### E. *Security*

In many applications, the embedding procedure must be secure in that an unauthorized user must not be able to detect the presence of embedded data, let alone remove the embedded data. Security requirements vary with application. The most stringent requirements arise in covert communication scenarios. The security of data-embedding procedures is interpreted in the same way as the security of encryption techniques. A secure data-embedding procedure cannot be broken unless the unauthorized user has access to a secret key that controls the insertion of the data in the host signal. Hence, a data-embedding scheme is truly secure if knowing the exact algorithm for embedding the data does not help an unauthorized party to detect the presence of embedded data. An unauthorized user should also be unable to extract the data in a reasonable amount of time even if

[2] See http://altern.org/watermark.

he knows that the host signal contains data and is familiar with the exact algorithm for embedding the data. Note that in some applications, e.g., covert communications, the data may also be encrypted prior to insertion in a host signal.

### *F. Copyright Protection and Ownership Deadlock*

Data-embedding algorithms may be used to establish ownership and distribution of data. In fact, this is the application of data embedding or watermarking that has received most attention in the literature. Unfortunately, most current watermarking schemes are unable to resolve rightful ownership of digital data when multiple ownership claims are made, i.e., when a deadlock problem arises. The inability of many data-embedding algorithms to deal with deadlock, first described by Craver *et al.* [15], is independent of how the watermark is inserted in the multimedia data or how robust it is to various types of modifications.

Today, no scheme can unambiguously determine ownership of a given multimedia signal if it does not use an original or other copy in the detection process to at least construct the watermark to be detected. A pirate can simply add his watermark to the watermarked data or counterfeit a watermark that correlates well or is detected in the contested signal. Current data-embedding schemes used as copyright-protection algorithms are unable to establish who watermarked the data first. Furthermore, none of the current data-embedding schemes has been proven to be immune to counterfeiting watermarks that will correlate well with a given signal as long as the watermark is not restricted to depend partially in a noninvertible manner on the signal.

If the detection scheme can make use of the original to construct the watermark, then it may be possible to establish unambiguous ownership of the data regardless of whether the detection scheme subtracts the original from the signal under consideration prior to watermark detection or not. Specifically, [16] derives a set of sufficient conditions that watermarks and watermarking schemes must satisfy to provide unambiguous proof of ownership. For example, one can use watermarks derived from pseudorandom sequences that depend on the signal and the author. Reference [16] establishes that this will work for *all* watermarking procedures regardless of whether they subtract the original from the signal under consideration prior to watermark detection or not. Reference [85] independently derived a similar result for a restricted class of watermarking techniques that rely on subtracting a signal derived from the original from the signal under consideration prior to watermark detection. The signal-dependent key also helps to thwart the "mix-and-match" attack described in [16].

An author can construct a watermark that depends on the signal and the author and provides unambiguous proof of ownership as follows. The author has two random keys $x_1$ and $x_2$ (i.e., seeds) from which a pseudorandom sequence $y$ can be generated using a suitable pseudorandom sequence generator [76]. Popular generators include RSA, Rabin, Blum/Micali, and Blum/Blum/Shub [25]. With the two proper keys, the watermark may be extracted. Without the two keys, the data hidden in the signal are statistically undetectable and impossible to recover. Note that classical maximal length pseudonoise sequences (i.e., $m$-sequence) generated by linear feedback shift registers are *not* used to generate a watermark. Sequences generated by shift registers are cryptographically insecure: one can solve for the feedback pattern (i.e., the keys) given a small number of output bits $y$.

The noise-like sequence $y$ may be used to derive the actual watermark hidden into the signal or to control the operation of the watermarking algorithm, e.g., to determine the location of pixels that may be modified. The key $x_1$ is *author* dependent. The key $x_2$ is *signal* dependent. The key $x_1$ is the secret key assigned to (or chosen by) the author. The key $x_2$ is *computed from the signal* that the author wishes to watermark. It is computed from the signal using a one-way hash function. For example, the tolerable error levels supplied by masking models (see Section IV) are hashed in [85] to a key $x_2$. Any one of a number of well-known secure one-way hash functions may be used to compute $x_2$, including RSA, MD4 [77], and SHA [60]. For example, the Blum/Blum/Shub pseudorandom generator uses the one-way function $y = g_n(x) = x^2 \bmod n$, where $n = pq$ for primes $p$ and $q$ so that $p = q = 3 \bmod 4$. It can be shown that generating $x$ or $y$ from partial knowledge of $y$ is *computationally infeasible* for the Blum/Blum/Shub generator.

The signal-dependent key $x_2$ makes counterfeiting very difficult. The pirate can only provide key $x_1$ to the arbitrator. Key $x_2$ is automatically computed by the watermarking algorithm from the original signal. As it is computationally infeasible to invert the one-way hash function, the pirate is unable to fabricate a counterfeit original that generates a desired or predetermined watermark.

Deadlock may also be resolved using the dual watermarking scheme of [85]. That scheme employs a *pair* of watermarks. One watermarking procedure requires the original data set for watermark detection. The second watermarking procedure does *not* require the original data set. A data-embedding technique that satisfies the restrictions outlined in [16] can be used to insert the second watermark.

The above discussion clearly highlights the limitation of watermarking as an unambiguous mean of establishing ownership. Future clever attacks may show that the schemes described in [16] or [85] are still vulnerable to deadlock. Furthermore, all parties would need to use watermarking techniques that have been proven or certified to be immune to deadlock to establish ownership of media. Note also that contentions of ownership can occur in too many different forms. Copyright protection will probably not be resolved exclusively by one group or even the entire technical community since it involves too many legal issues, including the very definition of similarity and derived works. Many multidisciplinary efforts are currently investigating standards and rules for national and international copyright protection and enforcement in the digital age.

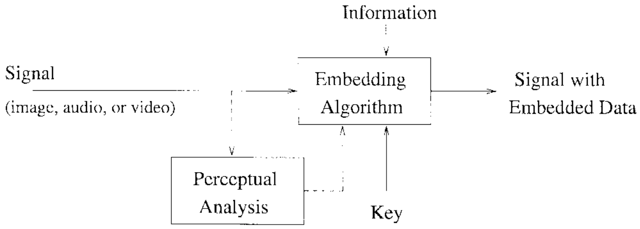


**Fig. 1.** Diagram of a data-embedding algorithm. The information is embedded into the signal using the embedding algorithm and a key. The dashed lines indicate that the algorithm may directly exploit perceptual analysis to embed information.

## IV. Signal Insertion: The Role of Masking

The first problem that all data-embedding and watermarking schemes need to address is that of inserting data in the digital signal without deteriorating its perceptual quality. Of course, we must be able to retrieve the data from the edited host signal, i.e., the insertion method must also be invertible. Since the data-insertion and data-recovery procedures are intimately related, the insertion scheme must take into account the requirement of the data-embedding application. In many applications, we will need to be able to retrieve the data even when the host signal has undergone modifications, such as compression, editing, or translation between formats, including A/D and D/A conversions.

Data insertion is possible because the digital medium is ultimately consumed by a human. The human hearing and visual systems are imperfect detectors. Audio and visual signals must have a minimum intensity or contrast level before they can be detected by a human. These minimum levels depend on the spatial, temporal, and frequency characteristics of the human auditory and visual systems. Further, the human hearing and visual systems are characterized by an important phenomenon called masking. Masking refers to the fact that a component in a given audio or visual signal may become imperceptible in the presence of another signal called the masker. Most signal-coding techniques (e.g., [41]) exploit the characteristics of the human auditory and visual systems directly or indirectly. Likewise, all data-embedding techniques exploit the characteristics of the human auditory and visual systems implicitly or explicitly (see Fig. 1). In fact, embedding data would not be possible without the limitations of the human visual and auditory systems. For example, it is not possible to modify a binary stream that represents programs or numbers that will be interpreted by a computer. The modification would directly and adversely affect the output of the computer.

### *A. The Human Auditory System (HAS)*

Audio masking is the effect by which a faint but audible sound becomes inaudible in the presence of another louder audible sound, i.e., the masker [42]. The masking effect depends on the spectral and temporal characteristics of both the masked signal and the masker.

Frequency masking refers to masking between frequency components in the audio signal. If two signals that occur simultaneously are close together in frequency, the stronger masking signal may make the weaker signal inaudible. The masking threshold of a masker depends on the frequency, sound pressure level, and tone-like or noise-like characteristics of both the masker and the masked signal [61]. It is easier for a broad-band noise to mask a tonal signal than for a tonal signal to mask out a broad-band noise. Moreover, higher frequency signals are more easily masked.

The human ear acts as a frequency analyzer and can detect sounds with frequencies that vary from 10 to 20 000 Hz. The HAS can be modeled by a set of bandpass filters with bandwidths that increase with increasing frequency. The bands are known as the critical bands. The critical bands are defined around a center frequency in which the noise bandwidth is increased until there is a just noticeable difference in the tone at the center frequency. Thus, if a faint tone lies in the critical band of a louder tone, the faint tone will not be perceptible.

Frequency-masking models are readily obtained from the current generation of high-quality audio codecs, e.g., the masking model defined in the International Standards Organization (ISO)-MPEG Audio Psychoacoustic Model 1 for Layer I [40]. The Layer I masking method is summarized as follows for a 32-kHz sampling rate. The MPEG model also supports sampling rates of 44.1 and 48 kHz.

The frequency mask is computed on localized segments (or windows) of the audio signal. The first step consists of computing the power spectrum of a short window (512 or 1024 samples) of the audio signal. Tonal (sinusoidal) and nontonal (noisy) components in the spectrum are identified because their masking models are different. A tonal component is a local maximum of the spectrum. The auditory system behaves as a bank of bandpass filters, with continuously overlapping center frequencies. These "auditory filters" can be approximated by rectangular filters with critical bandwidth increasing with frequency. In this model, the audible band is therefore divided into 24 nonregular critical bands.

Next, components below the absolute hearing threshold and tonal components separated by less than 0.5 Barks are removed. The final step consists of computing individual and global masking thresholds. The frequency axis is discretized according to hearing sensitivity and express frequencies in Barks. Note that hearing sensitivity is higher at low frequencies. The resulting masking curves are almost linear and depend on a masking index different for tonal and nontonal components. They are characterized by different lower and upper slopes depending on the distance between the masked and the masking component. We use $f_1$ to denote the set of frequencies present in the test signal. The global masking threshold for each frequency $f_2$ takes into account the absolute hearing threshold $S_a$ and the masking curves $P_2$ of the $N_t$ tonal components and $N_n$ nontonal components

$$S_m(f_2) = 10 * \log_{10}\Bigg[10^{S_a(f_2)/10} + \sum_{j=1}^{N_t} 10^{P_2(f_2,f_1,P_1)/10} + \sum_{j=1}^{N_n} 10^{P_2(f_2,f_1,P_1)/10}\Bigg]. \qquad (1)$$

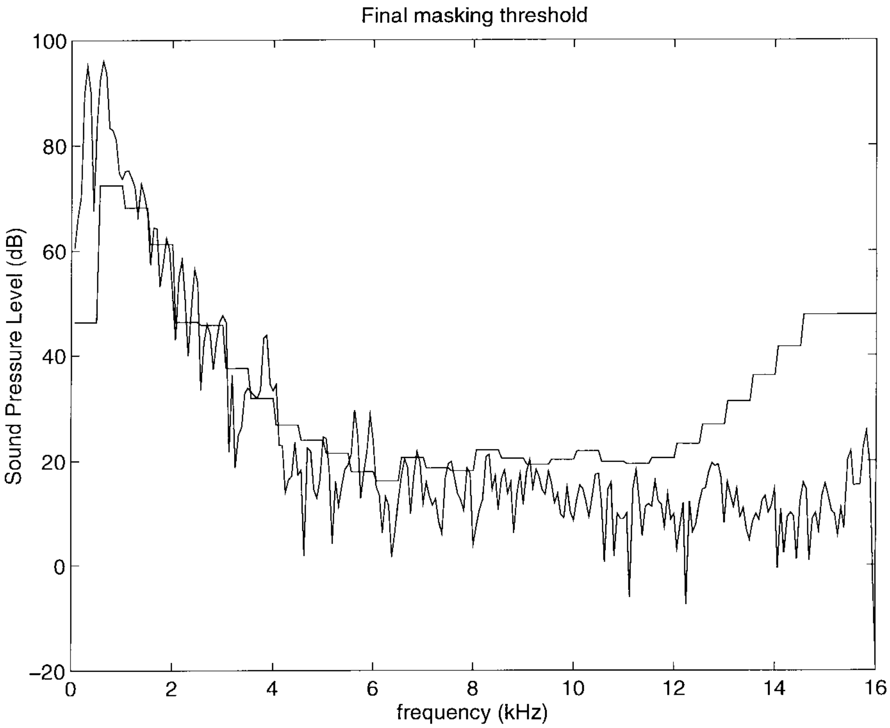


**Fig. 2.** Example of frequency masking in an audio signal. The original spectrum of the signal, along with the corresponding masking threshold, is shown in the plot.

The masking threshold is then the minimum of the local masking threshold and the absolute hearing threshold in each of the 32 equal-width subbands of the spectrum. Any signal that falls below the masking threshold is inaudible. An example plot of an original spectrum, along with the masking threshold, is shown in Fig. 2.

Temporal masking refers to both pre- and post-masking. Pre-masking effects render weaker signals inaudible before the stronger masker is turned on, and post-masking effects render weaker signals inaudible after the stronger masker is turned off. Pre-masking occurs 5–20 ms before the masker is turned on while post-masking occurs from 50–200 ms after the masker is turned off [61]. Note that temporal and frequency masking effects have dual localization properties. Specifically, frequency-masking effects are localized in the frequency domain, while temporal-masking effects are localized in the time domain.

Temporal-masking effects may be estimated using the envelope of the host audio. The envelope is modeled as a decaying exponential. In particular, the estimated envelope $t(i)$ of signal $s(i)$ increases with $s(i)$ and decays as $e^{-\alpha}$. A 32-kHz audio signal, along with its estimated envelope, is shown in Fig. 3.

### *B. The Human Visual System (HVS)*

Visual masking, which works in a fashion similar to audio masking, refers to a situation where a signal raises the visual threshold for other signals around it. As in audio, a spatial sinusoidal pattern will lower the detectability of other sinusoidal patterns whose frequencies are close to that of the sinusoidal pattern [48]. This is referred to as frequency masking. Similarly, spatial patterns can affect the visibility of other features that are spatially close to them. For example, luminance edges and fine details reduce the visibility of other signals around them.

In our work, we have used a model for frequency masking that is directly based on measurements of the amounts by which the visual threshold for signal gratings around a masking frequency are raised due to a masking grating at that frequency [48]. In particular, a model we use [99], based on the discrete cosine transform (DCT), expresses the contrast threshold at frequency $f$ as a function of $f$, the masking frequency $f_m$, and the masking contrast $c_m$

$$c(f, f_m) = c_0(f) \cdot \mathrm{Max}\{1, [k(f/f_m)c_m]^\alpha\} \qquad (2)$$

where $c_0(f)$ is the detection threshold at frequency $f$. The weighting function $k(f/f_m)$ centered about $f/f_m = 1$ is shown in Fig. 4.

To find the contrast threshold $c(f)$ at a frequency $f$ in an image, we first use the DCT to transform the image into the frequency domain and find the contrast at each frequency. Then, we use a summation rule of the form $c(f) = [\sum_{f_m} c(f, f_m)^\beta]^{1/\beta}$. If the contrast error at $f$ is less than $c(f)$, the model predicts that the error is invisible to human eyes.

A spatial masking model based on the threshold vision model is proposed by Girod [22]. The model accurately predicts the masking effects near edges and in uniform background. Assuming that the modifications to the image are small, the upper channel of Girod's model can be linearized [99] to obtain the tolerable error level for each

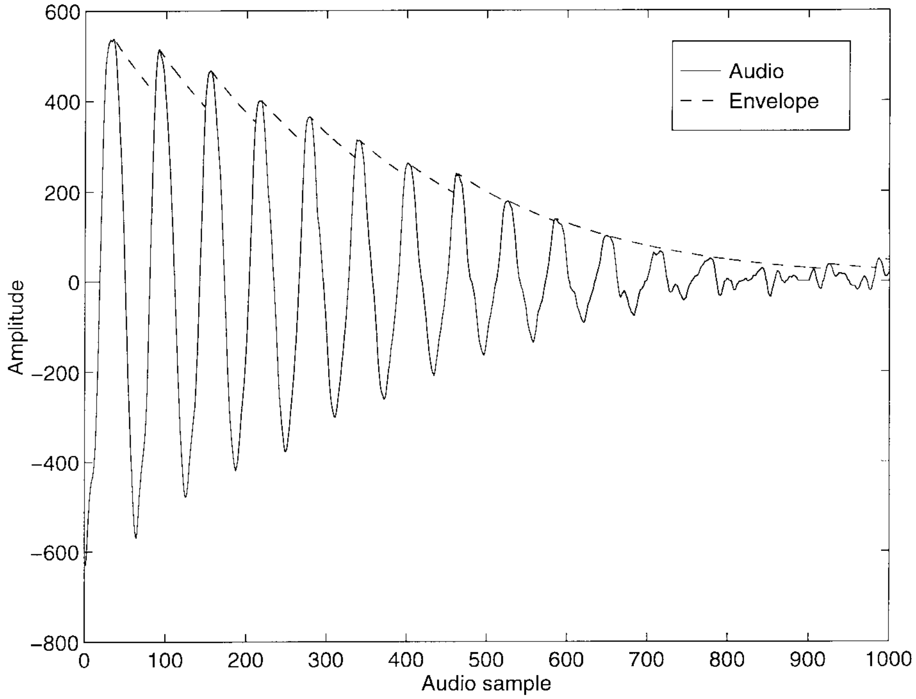


**Fig. 3.** Example of temporal masking in an audio signal. The original signal and the estimated envelope are plotted in the temporal domain.

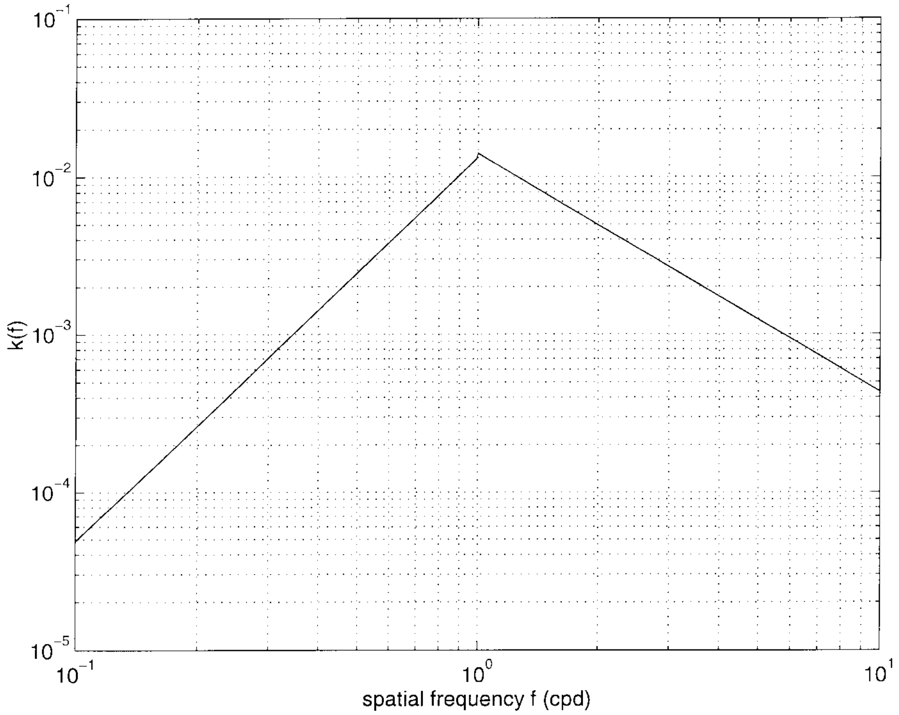


**Fig. 4.** Normalized image frequency-masking function.

coefficient. This is a reasonable assumption for transparent watermarking.

Under certain simplifying assumptions, the tolerable error level for a pixel $p(x,y)$ can be obtained by first computing the contrast saturation at $(x,y)$

$$dc_{\text{sat}}(x,y) = dc_{\text{sat}} = \sqrt{\frac{T}{\sum_{x',y'} w_4(0,0,x',y')}} \tag{3}$$

where the weight $w_4(x,y,x',y')$ is a Gaussian centered at the point $(x,y)$ and $T$ is a visual test-based threshold. Once $dc_{\text{sat}}(x,y)$ is computed, the luminance on the retina $l_{\text{ret}}$ is obtained from

$$dc_{\text{sat}}(x,y) = w_2(x,y) \cdot dl_{\text{ret}}(x,y). \tag{4}$$

From $dl_{\text{ret}}$, the tolerable error level $ds(x,y)$ for the pixel

$p(x,y)$ is computed from

$$dl_{\text{ret}}(x,y) = w_1(x,y) \cdot ds(x,y). \tag{5}$$

The weights $w_1(x,y)$ and $w_2(x,y)$ are based on Girod's model. The masking model predicts that changes to pixel $p(x,y)$ less than $ds(x,y)$ introduce no perceptible distortion.

## V. Image Data-Embedding Approaches

We begin with a review of image data-embedding techniques since they are the most common in the literature. Audio and video data-embedding algorithms are reviewed in the following sections. The data-embedding algorithms are classified into those that *implicitly* use masking and those that *explicitly* employ masking to embed the data. As described in Section IV, all data-embedding algorithms implicitly employ limitations of the HAS and HVS to embed data. However, some use advanced perceptual models to determine the best way to embed data.

### A. Image Data Embedding Implicitly Based on Masking

One of the simplest methods for inserting data into digital signals in noise-free environments is least significant bit (LSB) coding. The coding process begins with all of the LSB's of the host image set to zero (or all to one); zeroes and ones are then used to embed information in the LSB plane, e.g., a pattern or image in which "0" represents black and "1" represents white, or words coded in binary form. An analogous procedure can be used for color images, which are represented by three matrices for the intensities of the colors (e.g., red, green, and blue) in the image. LSB coding introduces noise of at most one unit, which, in practice, is imperceptible, so long as the host signal is not extremely low or weak.

LSB coding can be used to tag office memos onto digital data. For example, it has been used for the placement of markers to detect enlargements or reductions of an image that may have taken place during photo editing and to recover the associated dilation factor. Transparent cross marks are embedded in the LSB plane at fixed intervals in both the horizontal and vertical directions prior to editing. Changes in the dimensions made during editing can be detected and quantitatively measured by comparing the distances between the cross marks before and after the edit [27]. If cropping of an image is also expected, horizontal and vertical line numbers can be embedded at fixed intervals in the LSB plane to keep track of the pixel indexes from which a crop is made. The pixel index information will remain with the cropped image and can be recovered without a copy of the original, full-size image.

LSB coding has been practiced for decades; however, proposed use of the method for intellectual property-rights management is relatively new. The concept of a digital signature for authentication of electronic messages was introduced by Diffie and Hellman [19] and has since become a major area of research. More than a decade later, Matsui and Tanaka introduced the notion of video steganography in a series of conference papers that are surveyed in [54] and [55]. Embedding methods are proposed for grayscale, dithered binary, facsimile, and color still images and video.

**Table 1** Example of Cipher Key Table

| $\Delta_i$ | ... | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $c_i$ | ... | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | ... |

The first embedding scheme is for digitized grayscale image data, which consists of a set of integers between 0 and 255, representing the gray levels of an image at sampled points. The digitized image data $\{x_i\}$; $i \in N$ is converted to a sequence in which the first element is $x_1$ and subsequent elements are the differences between successive points, i.e., $\Delta_i = x_i - x_{i-1}$. Next, the person(s) embedding and extracting the message agree on the use of a particular cipher key table, which assigns a value $c_i$, either zero or one, to each $\Delta_i$ (see Table 1). To embed a binary sequence $B = \{b_i : b_i = 0 \text{ or } 1\}$; $i \in N$, look up the value of $c_i$ corresponding to $\Delta_i$ in the table. If $c_i = b_i$, then keep $\Delta_i$ as is. If $c_i \neq b_i$, go to the nearest $\Delta_j$ such that $c_j = b_i$ and substitute $\Delta_j$ in place of $\Delta_i$. The error introduced into the image data during the $i$th step is $\text{error}_i = \Delta_j - \Delta_i$, which is usually on the same order as noise, i.e., negligible. The hidden message can be retrieved by looking up the value for $c_i$ corresponding to $\Delta_i$.

In a second scheme, which uses ordered dithering, an image is divided into 4-by-4 pixel blocks and brightness thresholds $\{x_i\}$; $i = 0, 1, \cdots, 15$ for each of the 16 pixels in the block are assigned from top to bottom, left to right. Next, define sets

$$S_k = \{(x_i, x_j)_k : x_j - x_i = k\}, \qquad i, j = 0, 1, \cdots, 15; \\ i \neq j. \tag{6}$$

Let $(y_i, y_j)_k$ be a pair of output signals that pass through $(x_i, x_j)_k$. Then $(y_i, y_j)_k$ is either $(0,0)$, $(1,0)$, $(0,1)$, or $(1,1)$, where 0 indicates that the pixel will be turned "off" and 1 indicates "on." Only the pairs $(1,0)$ or $(0,1)$ will be used to embed a sequence of bits $B = \{b_n : b_n = 0 \text{ or } 1\}$; $n \in N$. To embed $b_n = 0$, set $(y_i, y_j)_k = (0,1)$. To embed $b_n = 1$, set $(y_i, y_j)_k = (1,0)$. To decode, disregard the $(0,0)$ and $(1,1)$ outputs and simply reverse the procedure described above.

Facsimile document signals serve as the host medium for a third message-embedding scheme. Documents are digitized following the international standard facsimile scanning rate of 8.23 pixels/mm in the horizontal direction [12].[3] The scanned data indicate whether a pixel is black or white, the two options. The message-embedding scheme is based on the fact that the data will be compressed using *run-length coding* (RLC) and modified *Huffman* coding schemes. RLC reduces data by replacing repeated characters with three characters: a *flag* character to signal that compression follows, the repeated character, and the number of repetitions. A binary message $B = \{b_n : b_n = 0 \text{ or } 1\}$; $n \in Z$ is embedded by shortening or lengthening runs by one pixel at the boundaries of the runs. In a simple illustrative example,

[3] See http://www.infomedia.net/scan/.

runs are set to be even number length when $b_i = 0$ (by leaving it as is if it is already of even length and lengthening it by one if it is odd) and to an odd length when $b_i = 1$ (by leaving it as is if it is already of odd length and by shortening it by one if it is even). Runs used for coding must have a length greater than two.

Van Schydel *et al.* [90] also propose LSB coding methods that rely on $m$-sequences for transparently embedding messages with greater security than straightforward LSB coding. A binary sequence is mapped from $\{0, 1\}$ to $\{-1, 1\}$. In one method, the message is embedded in the LSB plane using $m$-sequences. The second is based on LSB addition. Some disadvantages of the method are: hostile parties will know that all pixels are changed by either $\pm 1$; knowledge of a string of repeating consecutive bits enables the recovery of the data; and embedding is performed without regard to the DCT coefficients, so there is no guarantee of robustness with respect to JPEG.

Wolfgang and Delp [96], [97] extend van Schyndel *et al.*'s work to two dimensions. Localization, i.e., detection of the presence of the data, is improved under the new scheme. Localization relies on use of the cross-correlation function $R_{XY}$ of two images $X$ and $Y$, defined as

$$R_{XY}(\alpha, \beta) = \sum_i \sum_j X(i, y) Y(i - \alpha, j - \beta). \tag{7}$$

Let $X$ be the original image, $W$ the watermark, $Y$ the watermarked image, and $Z$ a possible forgery. The test statistic defined as

$$\delta = R_{YW}(0, 0) - R_{ZW}(0, 0) \tag{8}$$

can often detect whether or not $Z$ is a forgery. If $Z$ and $Y$ are identical, then $\delta = 0$. In one implementation of their method, Wolfgang and Delp embed data consisting of changes by $\{0, 1\}$. In another, the changes are bipolar, i.e., $\{-1, 1\}$. The bipolar data are easier to detect using the test statistic $\delta$. Both types of marking are robust with respect to the test statistic under JPEG compression. Information regarding the preservation of the data is not given.

LSB coding can and should be used in contexts that do not require more sophisticated approaches; however, it is not robust enough for general distribution because binary sequences are embedded in a manner that requires perfect preservation of the signal for successful extraction of the hidden message; noisy transmission, filtering, cropping, color space conversion, or resampling would destroy the message. A more serious concern with LSB coding is the possibility of extraction of a binary message by hostile parties.

To alert the user to contamination or tampering of LSB-coded data, Walton [95] suggests using check sums [78]. To circumvent interception of an embedded message by hostile parties, Walton [95] and Matsui and Tanaka [55] recommend controlling the embedding locations through the use of keys, e.g., the use of a pseudorandom number generator to determine a pseudorandom walk on the image pixel plane [44], [53]. After a user-specified number of steps, say, $N$, a check digit for the pixel values at the $N$ preceding positions is embedded in the $(N+1)$st pixel along the random walk. This procedure is repeated many times. Users should double-check that the path of the random walk does not cross over itself during the embedding of the check sums since it could lead to false alarms of tampering. If the possible discovery of the pseudorandom sequence-generation mechanism by a hostile party is a consideration, variations that disguise the locations of the check sums can be developed to prevent tampering with the check sums themselves.

For color images, the basic check-sum scheme can be applied, straightforwardly, three times to the three color planes. More interesting variations that take advantage of the three dimensions from the three color planes can be developed. The basis set for representing the images, for example, can be changed from red-green-blue (RGB) to hue-lightness-saturation; the check sum is then calculated in the new coordinate system, and the check-sum digit is encoded in the original coordinate system. (For further details on standard bases for color-image representation and conversion factors, see [79].) Matsui and Tanaka's embedding methods, with and without check sums, are not robust with respect to cropping. In concluding our discussion on LSB coding, we remark that steganographic freeware for image marking is widely available over the Internet.

In Ohnishi and Matsui's Haar wavelet transform-based method, messages are embedded by adding or subtracting one from the transform coefficients of the image [64]. Like LSB coding, the technique is very fragile with respect to simple tampering, e.g., cropping.

Sanford *et al.* developed software (BMPEMBED, ver. 1.51 in the C programming language) for embedding data information into and extracting the information from color images in bitmap format [80]. The principles of the method are outlined in the algorithm for grayscale images. Embedding consists of two main steps. First, an image is analyzed to identify pairs of elements (i.e., pixel values) $d_i$ and $d_j$, which are within a "noise range" $N$

$$|d_i - d_j| = \epsilon \leq N \tag{9}$$

and such that $f(d_i)$ and $f(d_j)$, the frequency of occurrence of $d_i$ and $d_j$, are fairly large and within a tolerance $\delta$

$$|f(d_i) - f(d_j)| < \delta. \tag{10}$$

Binary information will be embedded by using the value $d_i$ to represent a zero and $d_j$ to represent a one (or vice versa). The main principles used in marking grayscale images are extended to mark color images. Identification of pairs of pixels within acceptable noise ranges is a more complicated process for color-image data. Embedding in JPEG and wavelet transform compressed images is accomplished through the use of the same technique in the DCT and wavelet transform domains. In the experiments described in the report, robustness with respect to (further) JPEG compression and restoration is not considered or tested. This embedding method suffers from other drawbacks, e.g., fragility with respect to cropping and multiple embeddings and (depending on the image) limited available space for embedding. Since a fairly substantial amount of work is required to analyze images to determine suitable embedding locations, real-time image retrieval may only be possible if the locations are determined ahead of time; use of predetermined, fixed embedding locations for a given image increases the ease in tampering and extraction of embedded information by hostile parties.

Bender *et al.* propose *texture block coding,* in which a block from a region with a random texture is copied and placed in a region with similar texture [2]. Extraction of the hidden block is easy. Slides of the original image and its opposite (in which each pixel $x_i$ is replaced by $256 - x_i$) are overlayed. As one slide is moved across the other, a solid black region will appear when the embedded block and the original region, from which it was taken, overlap. The method cannot be applied to all images and is not amenable to automatic embedding since images must be examined one by one by a human to determine whether suitable textured regions exist. The method is not robust to cropping. Although regular, two-dimensional shapes (e.g., solid circles, rectangles) can be readily embedded, the technique is not well suited for handling intricate designs and textual information. Furthermore, texture block coding is easy to detect since anyone can make slides of the original image and its opposite and extract the embedded block or region.

The same scientists also proposed *patchwork,* a statistical approach, in which a subset of the pixels in an image is divided into two distinct sets. The brightness of one set of pixels is shifted by a positive number, and those from the other set are shifted by the corresponding negative number [2]. Only a limited amount of information can be embedded in an image using this approach, even if the image is divided into several regions and a different number is embedded into each (i.e., one number per region). The inventors provide data that the recovery rate is 85% after JPEG compression, with quality parameter 75%, which would likely not stand up as credible evidence beyond a reasonable doubt in a court of law.

Pitas and Kaskalis use shifting in an approach that allows slightly more information to be embedded [69], [70]. A binary signature that consists of equal numbers of "zeros" and "ones" is embedded in an image by assigning pixels into one of two sets. The intensity levels of the pixels in one of the sets are altered. The intensity levels are not changed in the pixels in the other set. The method is limited to signature embedding and cannot be used for embedding text messages. According to the inventors, the degree of certainty can be as low as 84% and as high as 92%, which would likely not stand up as evidence in a court of law for copyright protection. In [94], toral automorphisms are used to chaotically mix binary logos or signatures, which are added to a secret region in the image.

In a similar method by Dautzenberg and Boland, images are partitioned into blocks, and the mean is shifted by one or zero to embed binary code [17]. The code does not necessarily have to consist of equal numbers of zeros and ones. The authors claim robustness to lossy image compression, photocopying, color scanning, and dithering. The method suffers from at least two major drawbacks. The amount of information that can be embedded depends on the number of blocks in the image. The method cannot be used to mark images that will be widely distributed using different marks for each intended recipient because comparison of large numbers of differently marked images will allow hostile parties to recover the original image.

Bruyndonckx *et al.* also use a block partitioning based method but use the change in the mean values of the luminance in blocks for embedding [9]. After an image is divided into blocks, the embedding order and locations can be found using a key. The blocks are classified into one of three types of luminance: hard, progressive, and noise contrast. The pixels are then assigned to zones, which add an extra level of security for embedding. Two categories $A$ and $B$ are created in each zone. Each pixel is assigned to a category based on its block and zone assignment. Embedding of a bit $b$ in a block is carried out by changing the differences in the mean in the luminance values of pixels in categories $A$ and $B$ and zones 1 and 2

$$\begin{aligned}
\text{if } b = 0:\quad & m^*_{1B} - m^*_{1A} = L \\
& m^*_{2B} - m^*_{2A} = L \\
\text{if } b = 1:\quad & m^*_{1A} - m^*_{1B} = L \\
& m^*_{2A} - m^*_{2B} = L
\end{aligned}$$

where $m^*_{1A}$, $m^*_{1B}$, $m^*_{2A}$, and $m^*_{2B}$ are the mean values after embedding and $L$ is the embedding level. To render the embedding as transparent as possible, the inventors require that the mean value in each zone be left unchanged. This requirement uniquely determines the values of $m^*_{1A}$, $m^*_{1B}$, $m^*_{2A}$, and $m^*_{2B}$ after embedding. To summarize, six parameters are used to embed information: the embedding level $L$, the categories grid size, the block size, the number of bits to be embedded, the location of the blocks, and the level of redundancies (with the option of error-detection codes if the message is short enough to allow a high level of redundant embedding). Robustness to JPEG depends strongly on the compression ratio, embedding level, and grid sizes. Redundant embedding with error-correcting codes is very effective in reducing the error in extracted messages. Depending on the amount of information to be embedded (which, in turn, determines if redundant embedding can be accommodated), the scheme may or may not be robust to cropping.

Langelaar *et al.* propose a block-based method in which embedding is carried out in the luminance-hue-saturation (YUV) domain [46]. The pixel values from 8 × 8 JPEG blocks of an image (or multiples of them) are converted to the YUV domain. After embedding binary information, the blocks are converted back to the RGB domain. If an edge-enhancement filter is applied to the luminance pixel values, the error rate for bit extraction after JPEG filtering is reduced significantly (from over 10% for JPEG quality factor 90% to well under 5%). The error rate is under 5%

for JPEG quality factor 100–60% in experiments described in the paper by Langelaar *et al.*

Koch and Zhao's data-embedding method, used in the product SysCop, is similar to direct sequence and frequency hopping spread-spectrum communications [20], [68], [92] and is compatible with JPEG compression quality parameter 50% [51]. An image is partitioned into 8-by-8 blocks, and eight coefficients in the block are used for marking. The blocks are pseudorandomly selected to minimize detection. Since the locations of the eight coefficients have been published, hostile parties can use the information to corrupt or destroy a message. Decoding by unauthorized parties is more difficult because of the pseudorandom sequence used for embedding. Cropping of images may lead to difficulties in extracting messages that were pseudorandomly embedded. And cropping along lines that cut through, rather than along, the 8-by-8 JPEG blocks may lead to an image that is not robust to JPEG compression. Langelaar *et al.* report that image degradation is visible in their implementation studies to assess Koch and Zhao's method, and results are shown in a paper [46]. A variation on Koch and Zhao's method for image authentication is presented by Schneider and Chang [11]. The technique alters transform coefficients to enforce a relationship between the coefficients.

Spread-spectrum embedding spreads the watermark over a larger number of frequency ranges, the idea being that the energy in any given frequency range will be undetectable. The watermark can still be checked since the spreading pattern is known. Another spread-spectrum noise technique is described by Smith and Comiskey [82], where the authors hide data by adding a fixed-amplitude pseudonoise sequence to the image.

Kutter *et al.* use amplitude modulation and a secret key to embed a signature in color images [52]. The signature bits are repeatedly embedded in the blue channel to ensure robustness. The blue channel is used because the HVS is relatively less sensitive in this color domain. A single bit $s$ is embedded in a pseudorandomly selected pixel $p = (i, j)$ in an image $I = \{R, G, B\}$ by modifying the blue channel $B$ by a fraction of the luminance $L$, i.e.,

$$B_{i,j} \leftarrow B_{i,j} + q(2s - 1)L_{i,j}. \tag{11}$$

Here, $L = 0.299R + 0.587G + 0.114B$ and $q$ is a constant that represents the strength of the signature. $q$ is selected to optimize robustness and invisibility. The embedded message is retrieved using prediction of the original value of the pixel $p = (i, j)$ based on a linear combination of its neighboring pixel values. More precisely, the prediction $\hat{B}_{i,j}$ is

$$\hat{B}_{i,j} = \frac{1}{4c}\left(\sum_{k=-c}^{c} B_{i+k,j} + \sum_{k=-c}^{c} B_{i,j+k} - 2B_{i,j}\right) \tag{12}$$

where $c$ is the size of the cross-shaped neighborhood. The embedded bit is computed to be the difference $\delta$ of the predicted and coded bit

$$\delta_{i,j} = B_{i,j} - \hat{B}_{i,j}. \tag{13}$$

To reduce the possibility of incorrect retrieval, the bit is embedded many times, and the computed differences are averaged. Extension to an $m$-bit signature is straightforward. The inventors claim and discuss the extent to which the algorithm is robust to translation, rotations, slight blurring, JPEG attack, and composition with other images.

Puate and Jordan use fractal compression analysis to embed a signature in an image [74]. In fractal analysis, similar patterns are identified in an image. Domain blocks $D_b$, which represent patterns that appear frequently, are noted. The intent is to identify a set of domain blocks that can be transformed (i.e., contracted, isometrically transformed, luminance scaled, and luminance shifted) to approximate blocks within an image. The goal is to cover the entire image (as much as possible) with the transformed domain blocks. In fractal compression, the reduced information consists of domain blocks, appropriate transformations on the blocks, and pointers to locations where the transformed blocks should be mapped. A binary signature, consisting of the set of bits $\{s_i\}$, is embedded in an image by varying the regions in which pattern-matching searches are performed. For a bit $s_i$, range blocks are pseudorandomly chosen to be somewhere in region $A$ if $s = 1$ and in region $B$ if $s = 0$. The inventors present data that show that their method is robust with respect to JPEG compression. If the image is blurred before JPEG compression, the results are not as good. The binary message can be encoded to increase protection from interception by unauthorized parties. Only a limited amount of binary code can be embedded using this method. Since fractal analysis is computationally expensive, and some images do not have many large, self-similar patterns, the technique may not be suitable for general use.

Davern and Scott also propose a fractal-based steganographic method to embed binary messages, either plain or encrypted [18]. Fractal image analysis is used to identify similar blocks. Then transformations are constructed so that one similar block is transformed into an approximation for another. These transformations enable visibly imperceptible substitution of blocks. The set of blocks that can be used as substitutes is divided in two. To embed a zero, substitute blocks from the first set are used, and to embed a one, blocks from the second set are used. Davern and Scott's embedding method suffers from two of the same problems as that of Puate and Jordan. It is very slow, since fractal analysis is computationally expensive, and only a limited amount of information can be embedded. Additionally, Davern and Scott's method appears to be less robust to JPEG compression.

O'Ruanaidh *et al.* [65] describe a technique where image blocks are transformed using the DCT, Walsh transform, or wavelet transform. The data are embedded by incrementing a selected coefficient to encode a "1" and decrementing it to encode a "0." Coefficients are selected according to a criterion based on energy content. The watermark survived 20 : 1 JPEG image compression on the standard 256 × 256 Lena image. In a second approach [66], the authors describe a technique to embed information in the discrete Fourier transform phase.

Data-embedding techniques that require the original signal during detection are now reviewed. Some early algorithms for transparent marking, such as that by Cox *et al.* [14], require both the original and marked images for recovering an embedded message. The differences in the images is encrypted code. In the algorithm of Cox *et al.* [14], watermarks are embedded in the largest magnitude DCT coefficients to provide greater robustness to compression algorithms than LSB-type methods. Embedding in the highest coefficients corresponds to placing marks in the most perceptually significant portions of the image, regions that will remain relatively intact when subjected to compression. The marking algorithm consists of four steps.

- Compute the DCT and identify perceptually significant regions of the image suitable for watermark embedding.
- Construct the watermark $X = x_i, x_2, \cdots, x_n$, where each $x_i$ is chosen according to $N(0,1)$, where $N(\mu, \sigma^2)$ denotes a normal distribution with mean $\mu$ and variance $\sigma^2$.
- Insert the watermark in the DCT domain of the image by setting the frequency component $\nu_i$ in the original image to

  $$\nu_i \leftarrow \nu_i(1 + x_i \alpha_i) \tag{14}$$

  where $\alpha_i$ is a scalar factor.
- Compute the inverse DCT of the sum from the previous step to recover a transparently marked image.

Note that $n$, the number of DCT coefficients affected by the watermark, indicates the extent to which the watermark will be spread out among the components of the image. The authors choose $\alpha_i$ to be 0.1 in their experiments. A better approach would be to set $\alpha_i$ adaptively to different values for different frequencies. A Gaussian type of watermark is used for watermarking because it is much more robust to tampering than uniform embedding, particularly when $n$ is large. More specifically, the authors claim that $\Omega(\sqrt{n/\ln n})$ similar types of watermarks would have to be embedded to have “any chance” of destroying the image. The larger the $n$, the greater the protection provided by the watermark.

Extraction of the watermark by Cox *et al.* consists of four steps.

- Compute the DCT of a (possibly) watermarked image.
- Compute the DCT of the original image.
- Compute the difference in the results from the previous two steps to a watermark $X^*$.
- Compare the extracted mark $X^*$ with the original watermark $X$.

Comparison of the marks is conducted using a similarity measure defined by

$$\mathrm{sim}(X, X^*) = \frac{X^* \cdot X}{\sqrt{X^* \cdot X^*}}. \tag{15}$$

Studies on the robustness of the watermarks by Cox *et al.* show that when $n > 1000$ (i.e., 1000 perceptually significant frequency components of the image spectrum are altered), the watermark is reliable after JPEG encoding (quality factor 5%), dithering, clipping, clipping with JPEG encoding (quality factor 10%), and the combination of printing, photocopying, subsequent rescanning, and rescaling. When five watermarks were embedded in an image, all were successfully identified. Successful identification is also reported in averaging five separately watermarked images. Stone has published a report in which he shows how the strength of the technique proposed by Cox *et al.* can be diminished [83] when multiple watermarked copies are available to a pirate.

Ragusa *et al.* [75] devised and implemented a modification of the algorithm by Cox *et al.* in which regions of interest (ROI’s) are identified from the DCT components. Ragusa *et al.* assume that for most images, ROI’s that need to be watermarked for protection will have prominent edges. This assumption reduces the regions that will be marked so that only 200 or so ROI’s need to be marked, as opposed to the 1000 recommended by Cox *et al.* The primary advantage of the modified scheme is the reduced time required for embedding and extraction and a decrease in the likelihood of noticeable perceptual differences. The disadvantage is that solid regions and regions without prominent edges will not be marked.

Another algorithm that requires the original image is that by Hsu and Wu [36]. The signature is an image of a seal with Chinese characters. Permutations of the middle-band coefficients of the is DCT are used for encoding. The difference between the original and marked image is used to retrieve the signature. Pseudorandom number generators can be used to increase the security. They can serve several roles, such as designating the permutation and the embedding location. The inventors claim that the signed images are robust to general image-enhancement techniques and JPEG lossy compression.

In Fridrich [21], a low-frequency-based scheme similar to that of Cox *et al.* is introduced. In the scheme, a random black-and-white pattern is processed by a cellular automaton with the voting rule through several stages and smoothed with a low-pass kernel. The resulting pattern is added to the image. The robustness of the technique is reportedly slightly better than [14].

In an approach, called *tagging,* by Caronni [10], images are partitioned into blocks, and the mean values of the brightness of pseudorandomly selected blocks are altered to embed code. (When very high security is required, better random position-selection mechanisms should be used.) The inventor claims that the tags can be recovered after JPEG compression with quality parameter 30%. And when a modulation strength of 2% and tag size of 16 × 16 pixels are used, 75% of the tags from enlarged, color-printed, and rescanned images can be recovered. In most cases, tags with a modulation strength of 2% are imperceptible. Let $b_0(x,y)$ and $b_m(x,y)$ represent the brightness of the original and tagged images and $m_0$ and $m_m$ represent their mean. Then the covariance $\nu_{0m}$ between and variances $\nu_0$, $\nu_m$ of the original and marked image are

$$\begin{aligned}\nu_{0m} &= c\sum_{x=1}^{X}\sum_{y=1}^{Y}(b_0(x,y)-m_0)(b_m(x,y)-m_m)\\ \nu_0^2 &= c\sum_{x=1}^{X}\sum_{y=1}^{Y}(b_0(x,y)-m_0)\\ \nu_m^2 &= c\sum_{x=1}^{X}\sum_{y=1}^{Y}(b_m(x,y)-m_m)\end{aligned}\tag{16}$$

where $c = 1/(X * Y - 1)$. If two images are identical, then the correlation coefficient between the two images $||R|| = \nu_{0m}/\sqrt{\nu_0 \nu_m} = 1$. As the images become more dissimilar from tagging, $||R|| \rightarrow 0$. Note that only images of the same size can be compared using this method of measurement. The tag is recovered by comparing the original and marked image and the key for determining the order and positions of the tagged blocks.

Many commercial image data-embedding algorithms are available. Digimarc's PictureMarc[4] uses a spread-spectrum technique. Using a 32-bit binary signature, the least significant bit of the 32 is aligned with the first pixel of the image. If the bit is a "1," the first random number is added to the first pixel; if the bit is a "0," then it is subtracted. The same process is applied to the second pixel using the second bit of the signature number to choose whether to add or subtract the second random number. This is continued until all 32 bits have been used. The process is then repeated by starting over at bit 0 while continuing across the image. The algorithm is terminated when all pixels have been marked. If the original is available during detection, it is subtracted from the signed image, resulting in a difference signal. The sign of each pixel in the difference image is compared with the corresponding code pattern sign. If they match, then the identification signal bit is a one; otherwise, it is a zero. This process is repeated over the entire image, and the individual bits are summed. The sum of each bit is divided by the number of repetitions of the signature. The result is the identification number. In the cases where there is no original image or changes have occurred to the signed image, small-signal detection methods are used to read the signature.

Other commercial software include DICE's Argent technology[5] and Aliroo's ScarLet algorithm.[1]

### B. *Image Data Embedding Explicitly Based on Masking*

A technique based on spatial and frequency masking is presented in [84]. The data embedding works by breaking an image into small (e.g., 8 × 8) blocks. Each block is then represented as a vector $v$ in an $n$-dimensional space (e.g., $n = 8^2$) and projected onto a normalized pseudorandom (author-defined) direction $z$ weighted by the masking values for the particular block. With $z$ normalized, the projection of the image block onto the user-defined direction is simply the inner product $p = \langle v, z\rangle = \sum_i v(i)z(i)$.

**Fig. 5.** Original 512 × 512 grayscale image.

The scalar projection value $p$ is then quantized with respect to the masking levels of that block $T$, creating the value $p^*$. The quantized value is perturbed by $\pm\frac{1}{4}T$ to embed the data, i.e., $p' = p^* \pm \frac{1}{4}T$. The new projection $p'$ contains the hidden data. To extract the hidden data, each recovered block is projected onto the appropriate pseudorandom direction, and a simple remainder operation is applied

$$b = \begin{cases}1, & \text{if } (\langle v', z\rangle - [\langle v', z\rangle]) > 0;\\ 0, & \text{otherwise}\end{cases}\tag{17}$$

where $[\cdot]$ is the rounding operation and $b$ is the bit embedded in the block. The technique easily accommodates the insertion of multiple bits per block. Figs. 5 and 6 provide an example of the algorithm. The image in Fig. 6 was obtained by embedding an 8192-bit text file inside the original image shown in Fig. 5. Note that the two images are perceptually identical. The bit error rate for the embedded data after different levels of JPEG coding is shown in Fig. 7. Using error correction codes and bit repetition, the data are able to survive low JPEG qualities. For example, one may use 15 : 1 bit repetition (reducing the effective embedded bit rate to about 500 bits) for environments with a lot of distortion.

Several masking-based image data-embedding algorithms that require the original during detection have been proposed. Cox *et al.* [14] propose to set $\alpha_i$ (14) adaptively according to masking values for different frequencies.

A block-based version of [14] that employs visual models has been developed by Podilchuk and Zeng [71]. The algorithm employs the just noticeable difference paradigm employed by perceptual coders. The watermarked DCT coefficients $X^*(u, v, b)$ are generated by (18), as shown at the bottom of the next page, where $X(u, v, b)$ refers to the DCT coefficients at location $(u, v)$ in block $b$ of the image, $w(u, v, b)$ is the sequence of real valued watermark values,

[4] See http://www.digimarc.com.

[5] See http://www.digital-watermark.com:80/.

**Fig. 6.** Image with embedded information.

and $J(u, v, b)$ is the computed just noticeable difference calculated from the visual models. The detection is performed in the same manner as that of Cox *et al.* The authors claim that the watermark survives uniform noise, cropping, and JPEG compression with a quality factor of 20, as well as printing, photocopying, rescanning, and rescaling of the image. The results were slightly better than the original Cox *et al.* algorithm, indicating some of the advantages of employing masking for data embedding. Another efficient image data-hiding algorithm based on perceptual masking is presented in [24].

In [88], the authors present a technique that exploits HVS to guarantee that the embedded watermark is imperceptible and robust. The watermark is generated by filtering a pseudorandom sequence [owner identification (id)] with a filter that approximates the frequency and spatial masking characteristics of the HVS. The watermark is generated by segmenting the image into blocks $B_{i,j}$ of size $n \times m$, e.g., 8 × 8. For each block $B_{i,j}$, there are the following steps.

1) Compute the DCT $D_{i,j}$ of the image block $B_{i,j}$.

2) Compute the frequency mask $M_{i,j}$ of the DCT image block $D_{i,j}$.

3) Use the mask $M_{i,j}$ to weight the noise-like author id for that image block, creating the shaped author signature $P_{i,j}$.

4) Create the watermark block $W_{i,j}$ by computing the inverse DCT of $P_{i,j}$ and weighting the result with the corresponding spatial mask $S_{i,j}$.

5) Add the watermark $W_{i,j}$ to the block $B_{i,j}$, creating the watermarked block $B'_{i,j}$.

Detection of the watermark is accomplished via hypothesis testing [91] using the similarity measure defined by (15).

To illustrate the algorithm, the technique was applied to the original image shown in Fig. 8(a). The resulting watermarked image is shown in Fig. 8(b), along with the watermark shown in Fig. 8(c). The watermark has been rescaled to gray levels for display. The watermark values corresponding to smoother background regions are generally smaller than watermark values near edge regions. This is to be expected, as edge regions have more favorable masking characteristics. The absolute values of the watermark range from 2 (smooth regions) to 48 (edge regions).

The robustness of the algorithm to JPEG coding is shown in Fig. 9. The plot indicates the similarity values of the "Peppers" test image with and without a watermark at several bit rates corresponding to JPEG qualities from 5 to 95%. To simulate additional attacks on the watermark, colored noise was added to the test image *prior* to JPEG coding. Each coding quality was tested 100 times, with a different colored noise sequence used during each test. The error bars at each quality correspond to the maximum and minimum similarity values at each bit rate. Even at very low image quality, the similarity values are separated, allowing the existence of a watermark to be easily determined.

## VI. Audio Data-Embedding Approaches

### A. Audio Data Embedding Implicitly Based on Masking

Several techniques have been proposed in [2] and [28]. Using a phase-coding approach, data are embedded by modifying the phase values of Fourier transform coefficients of audio segments. The authors also proposed embedding data as spread-spectrum noise. A third technique, echo coding, employs multiple decaying echoes to place a peak in the cepstrum at a known location.

Another audio data-embedding technique is proposed in [89], where Fourier transform coefficients over the middle frequency bands, 2.4–6.4 kHz, are replaced with spectral components from a signature. The middle frequency band was selected so that the data remain outside of the more sensitive low-frequency range. The signature is of short time duration and has a low amplitude relative to the local audio signal. The technique is described as robust to noise and the wow and flutter of analog tapes.

Pruess *et al.* [73] embed data into audio by shaping a pseudonoise sequence according to the shape of the original signal. The data are embedded within a preselected band of the audio spectrum after proportionally shaping it by the corresponding audio-signal frequency components. In

$$X^*(u,v,b) = \begin{cases} X(u,v,b) + J(u,v,b)w(u,v,b), & \text{if } X(u,v,b) > J(u,v,b); \\ X(u,v,b), & \text{otherwise} \end{cases} \tag{18}$$

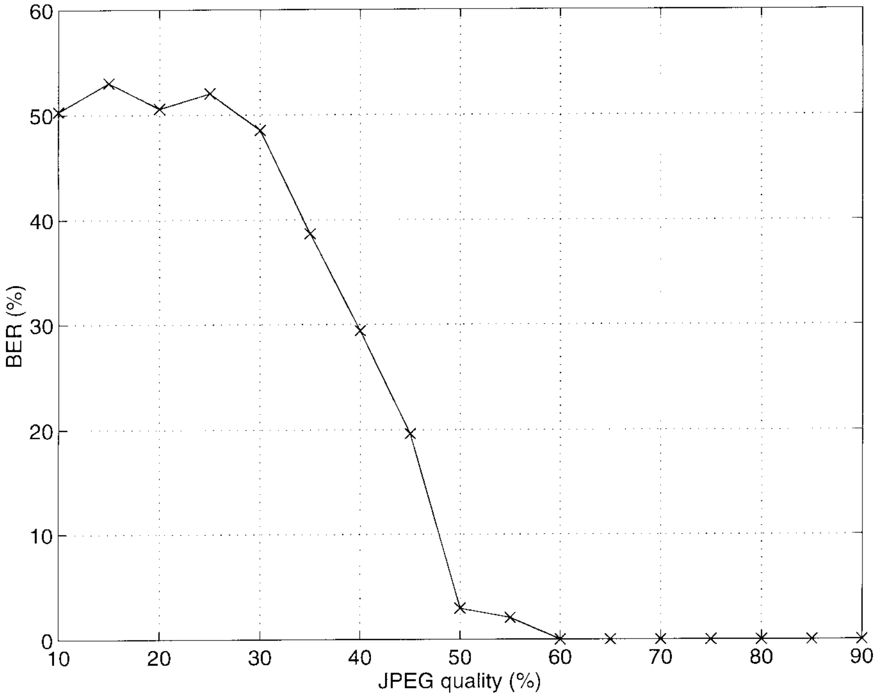


**Fig. 7.** Bit error rate versus JPEG quality factor for image shown in Fig. 6.

particular, the frequency component $\nu_i$ in the original audio signal is modified to

$$\nu_i \leftarrow \nu_i(1 + x_i\alpha_i) \tag{19}$$

where $x_i$ is a sample of the spread data and $\alpha_i$ is a scalar factor. The inventors claim the composite audio signal is not readily distinguishable from the original audio signal. The data may be recovered by essentially reversing the embedding operation using a whitening filter. As described above, a very similar embedding technique was later employed by Cox *et al.* for image watermarking.

Some commercial products are also available. The Identification Code Embedded system from Central Research Laboratories inserts a pair of very short tone sequences into an audio track.

Solana Technology Development Corporation [47] embeds data into subbands of the audio signal. The data to be embedded modulate a pseudonoise spread-spectrum signal, each subband of which has a bandwidth corresponding to those of the digital audio signal. The modulated data carrier sequence is combined with the audio subband samples to form a combined signal in which the embedded data are carried. The combined signal is then combined into the audio signal. At the decoder, the combined signal is demodulated to recover the auxiliary data signal. The recovered auxiliary data signal is inaudible in the audio signal and is spectrally shaped according to the audio signal to enhance concealment. Solana has an audio marking product called Electronic DNA (E-DNA) and ScarLet by Aliroo.

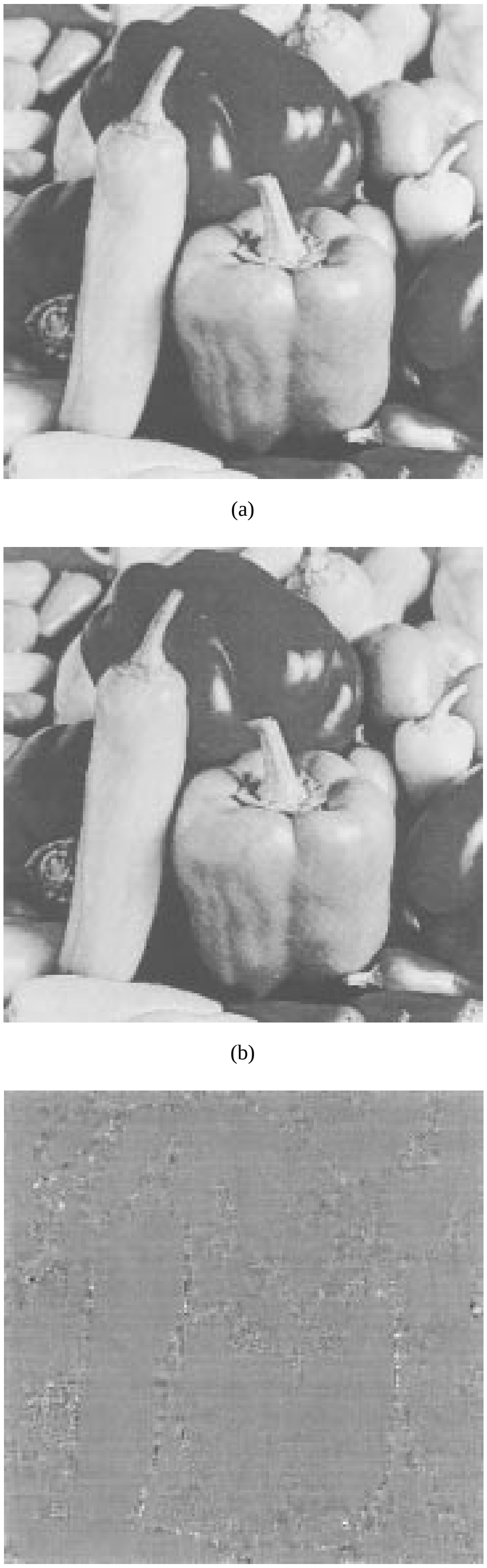
Patents for audio data embedding have been filed by Radio Audit Systems, Inc., for a radio-broadcast signal-processing system with an information encoder [26] and the DICE Company for an apparatus and method for encoding information into digital multimedia data [13].

Very few audio data-embedding algorithms that use the original audio signal during detection have been proposed. A few image watermarking schemes, e.g., [14], have been described as generic and applicable to audio, although no results have been reported. This is due to the fact that most audio embedding algorithms are designed for broadcast environments. As a result, most audio embedding algorithms are required to retrieve the embedded information without access to the original.

### *B. Audio Data Embedding Explicitly Based on Masking*

An audio data-embedding technique based on temporal and frequency masking is presented in [84]. The data embedding works by extracting length 512 blocks of the audio. Sequential blocks of length 512 are extracted from the audio signal and projected onto a pseudorandom (author-defined) direction weighted by the masking values for the particular block. The scalar projection value is then quantized with respect to the masking levels of that block. To embed data, the quantized projection value is perturbed to a new value. To extract the hidden data, each recovered block is projected onto the appropriate pseudorandom direction and a simple remainder operation is applied [see (17)].

Moses [57] proposes a technique to embed data by encoding it as one or more whitened direct sequence spread-spectrum signals and/or a narrow-band frequency-shift-keying data signal and transmitted at the time, frequency, and level determined by a neural network (NN) such that the signal is masked by the audio signal. The NN monitors the audio channel to determine opportunities to insert the data signal such that the inserted signals are masked.



(a)

(b)

(c)

**Fig. 8.** An image (a) original and (b) watermarked. (c) Watermark rescaled to gray levels for display.

In [4] and [87], the authors present an audio watermarking algorithm that exploits temporal and frequency masking (see Section IV-A) to embed robust and inaudible data. The watermark is constructed by breaking each audio clip into smaller segments and adding a perceptually shaped pseudorandom sequence.

An example showing the robustness of the watermarking technique to MPEG coding is shown in Fig. 10. The audio signal is the beginning of the third movement of the sonata in B flat major D 960 of Schubert (piano, duration 12.8 s), interpreted by Ashkenazy. The coding/decoding was performed using a software implementation of the ISO/MPEG-1 Audio Layer II coder with several different bit rates: 64, 96, and 128 kbits/s. The plot shows the similarity measure of the audio piece with and without the watermark. The increments on the $x$-axis correspond to 1.16-s segments of audio. For example, the similarity values for block number 2 are measured over the piano signal from $t = 1.16$ s to $t = 2.32$ s. As expected, the similarity values vary over time as the power of the watermark varies temporally with the power of the host signal. Observe that the upper similarity curve for the audio piece is widely separated from the lower curve over the entire duration of the signal.

## VII. Video Data-Embedding Approaches

### A. *Video Data Embedding Implicitly Based on Masking*

Fewer documents describing video data embedding are available in the public domain relative to image embedding. Most works are statements in papers to the effect that straightforward extension of a proposed still-image marking technique would be effective, e.g., [18] and [51]. For copyright protection, video data embedding must meet several requirements in addition to those for still images because the volume of data is of a much higher order and real-time embedding may be required in some applications, such as video-on-demand systems. The remainder of this section will focus on works that exploit the three-dimensional character of video data (i.e., two-dimensional images in the temporal domain) and characteristics associated with MPEG compression. Approaches that involve explicit computation of upper and lower bounds for masking by the HVS will then be described.

Video data embedding for copy protection has become a very pertinent issue. The Data Hiding Subgroup of the Copy Protection Technical Working Group is currently evaluating proposals for DVD protection. A data-embedding system is required to mark video content for the purposes of identifying marked material and preventing unauthorized recording/playback. The goal of the technologies is to allow content providers to mark all copyrighted digital video material (NTSC, PAL, and SECAM) with the watermark. Video recorders/players would respond appropriately by refusing to record or play improperly marked material.

One of the earliest examples of video data embedding was proposed by Matsui and Tanaka [55]. First, a frame from the video is divided into 8 × 8 blocks, and the

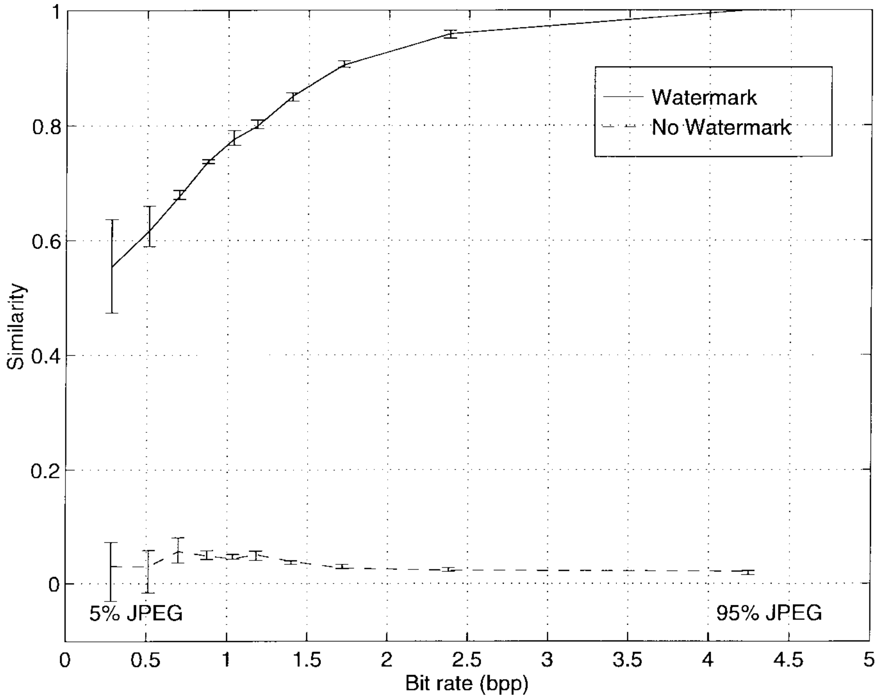


**Fig. 9.** Detection of watermark in Peppers image after JPEG coding, with qualities from 5 to 95%.

two-dimensional DCT is computed using the formula

$$p_{ij} = \frac{2K(i)}{\sqrt{N}} \cos \frac{(2j+1)i\pi}{2N} \tag{20}$$

where $i, j = 0, 1, 2, \cdots, 7$, $N = 8$ and

$$K(i) = \begin{cases} 1/\sqrt{2}, & i = 0, \\ 1, & i = 1, 2, \cdots, 7. \end{cases} \tag{21}$$

The coefficient matrix $A$ for the DCT is

$$\begin{aligned} A &= [a_{ij}] \\ &= [p_{ij}] \cdot [s_{ij}] \cdot [p_{ij}]^T \end{aligned} \tag{22}$$

where $i, j = 0, 1, 2, \cdots, 7$ and $s_{ij}$ is the pixel value of the $(i, j)$th element in the block. The matrix $A$ is quantized by

$$b_{ij} = \frac{a_{ij}}{\alpha \cdot t_{ij}} \tag{23}$$

where $i, j = 0, 1, 2, \cdots, 7$, $t_{ij}$ is the threshold factor, $\alpha$ is specified by a user to control the bit rate per pixel, and the values of $b_{ij}$ are rounded to the nearest integer. Information $M_k$ is embedded by converting it into a binary sequence $\{m_{kn}\}$, where $n = 1, 2, \cdots$ and $m_{kn} = 0$ or 1. If $m_{kn} = 0$, then $a_{ij}/(\alpha \cdot t_{ij})$ is set to be the nearest odd integer $b_{ij}$, and if $m_{kn} = 1$, then $a_{ij}/(\alpha \cdot t_{ij})$ is set to be the nearest even integer. We denote the operation by $f(b_{ij})$, the modified $b_{ij}$ by $b_{ij}^{(n)}$, where

$$b_{ij}^{(n)} = f(b_{ij}, m_{kn}) \tag{24}$$

and the matrix of $b(n)_{ij}$ as

$$B' = \left[ b_{ij}^{(n)} \right]. \tag{25}$$

This data-embedding method is very fragile with respect to noise, cropping, and repeatedly embedding data, since values are only changed incrementally up or down. Furthermore, if many copies of the same video frame with different embedded marks are available, hostile parties could compare them and might be able to determine the original, unmarked video frame. Embedded messages might be extracted by hostile parties unless supplementary precautions (e.g., message encryption prior to embedding, pseudorandom embedding locations) are used. Since robustness with respect to MPEG video compression was not a consideration during the design and test phase of the method, no data on the subject are given. It is unlikely that the embedded messages would be preserved after substantial MPEG video compression, but the level of fragility is not known.

Kinoshita *et al.* [50] have developed a method to embed information transparently in the MPEG motion vector in video. North Carolina State University's SHANG Group has developed software called "Secure MPEG," which is available for free over the Internet.[6] Sanford *et al.* are currently investigating extensions of their data-embedding method for still images to video data [80].

In extensive studies to extend concepts from still-image marking by Cox *et al.* [14], Hartung and Girod [23], [29]–[32] propose a spread-spectrum data-embedding method for raw (or uncompressed) video data and a second for MPEG bitstream data. The marking of raw video data $v_i$ to produce a modified signal $\hat{v}_i$ is described by

$$\hat{v}_i = v_i + \alpha \cdot b_i \cdot p_i \tag{26}$$

[6] See http://shang.csc.ncsu.edu/smpeg.html.

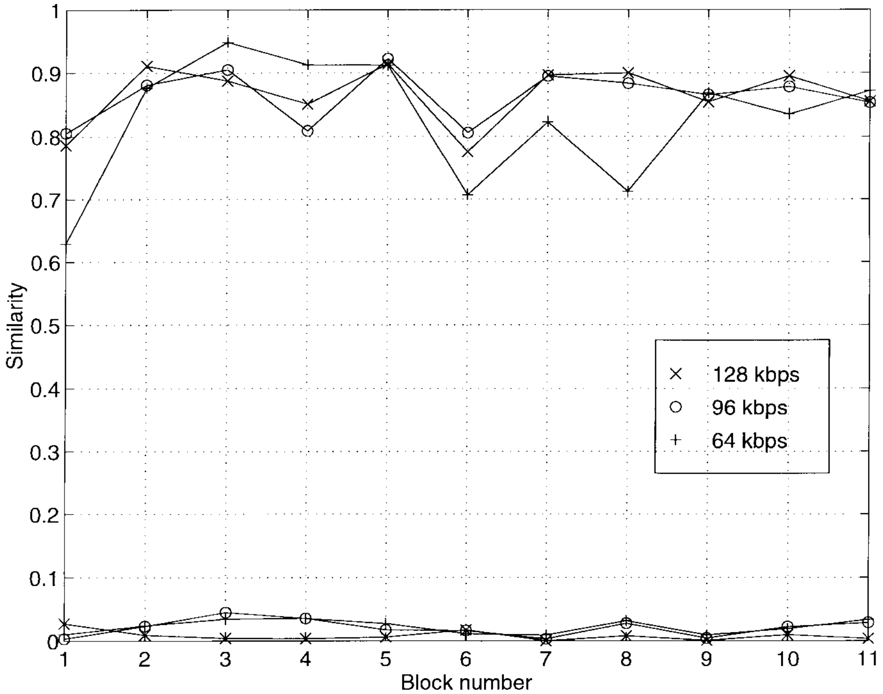


**Fig. 10.** Detection of watermark in piano piece after MPEG coding at 64, 96, and 128 kbits/s.

where $p_i$ is the pseudonoise sequence, $b_i$ is the embedded bit, and $\alpha$ is a scaling factor. The information is recovered using a matched filter. Use of a variable scaling factor (i.e., use of a sequence $\alpha_i$ rather than a constant $\alpha$) would improve the level of security and robustness. Similar to Cox *et al.*, the authors remark that $\alpha$ can be varied according to local properties of the video frame or to spatial and temporal masking properties associated with the HVS. A better receiver, which employs a prefilter before decorrelation, is presented in [32].

Hartung and Girod also have a data-embedding method for MPEG compressed video. The approach indirectly exploits masking characteristics, as it is based on the MPEG stream. One of its explicit goals is to avoid an increase in the bit rate. The embedding procedure consists of seven steps.

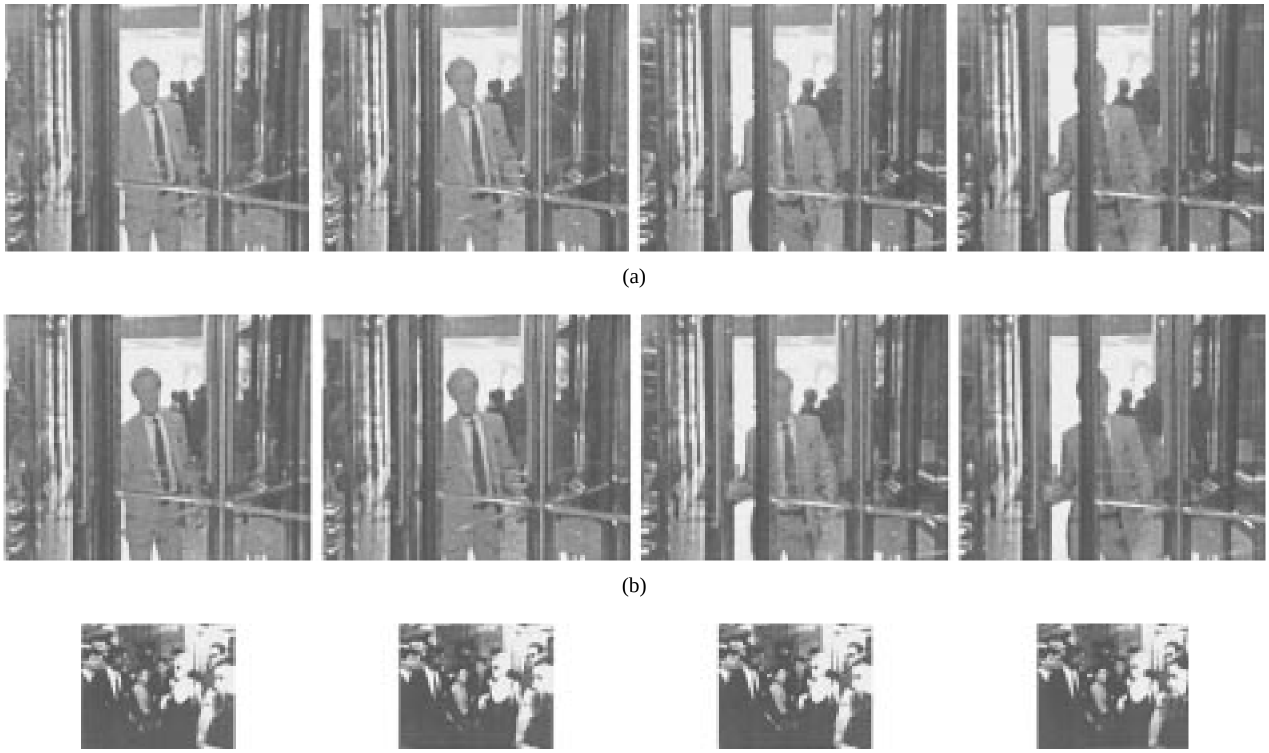
1) Compute the DCT of the data. Rescan the DCT coefficients using a zigzag scan to obtain a 1 × 64 vector. Let $W_0$ denote the zero-frequency (DC) coefficient and $W_{63}$ the highest frequency (AC) component.
2) Let $\hat{V}_n$ and $V_n$ denote the DCT coefficient of unmarked and marked signals and let $\hat{V}_0 = V_0 + W_0$.
3) Find the next variable-length coder (VLC) in the bitstream and identify the run-level pair $(r_m, l_m)$ belonging to the code word and the position and amplitude of the AC DCT coefficient $V_m$ represented by the VLC code word.
4) Let $\tilde{V}_m = V_m + W_m$ denote the candidate for new DCT coefficient. Check that the use of $\tilde{V}_m$ will not increase the bit rate.
5) Let $R$ and $\tilde{R}$ be the number of bits used for transmitting $(r_m, l_m)$ and $(r_m, \tilde{l}_m)$, respectively.
6) If $R \geq \tilde{R}$, then transmit $\tilde{R}$. Else transmit $(r_m, l_m)$.
7) Repeat steps 3)–6) until an end-of-block code word is encountered.

Hartung and Girod discuss the robustness of their method with respect to several types of attacks discussed in Section III-D. They remark that their data are robust to compression, filtering, and modest rotations. A detection and correction mechanism is needed for larger rotations. Removal and insertion of data lead to loss of synchronicity of the pseudonoise sequence between the sender and receiver so that a mechanism for detecting the loss and for resynchronizing the pseudonoise sequence is needed to thwart the attacks.

Similar to the situation with audio data-embedding algorithms, almost no video data-embedding algorithms that use the original have been proposed. Again, a few image watermarking algorithms, e.g., [14], have been described as generic and applicable to video.

### B. Video Data Embedding Explicitly Based on Masking

In [84], the authors present a projection-based video watermarking algorithm from an extension of their image data-embedding algorithm (see Section V). An example of the technique is shown in Fig. 11. In the example, a 311 frame, 120 × 160 grayscale video of a Madonna video is embedded in an equal-length sequence from the movie *Broadcast News*. The Madonna video is embedded for *real-time playback* along with the host video, i.e., 30 frames per second. The Madonna video is encoded using MPEG at a bit rate of 294 bytes per frame (8820 bytes per second). The frames of the *Broadcast News* video are of size 240 × 360. Sample frames from each of the videos are shown in Fig. 11.



(a)

(b)

(c)

**Fig. 11.** Video-in-video application. (a) Original *Broadcast News* video. (b) *Broadcast News* video with embedded (c) Madonna video. The Madonna video is embedded in real time.

In [86], the authors propose an object-based video watermarking technique. To address issues associated with video motion and redundancy, individual watermarks are created for objects within the video. Similar strategies were discussed at the July 1997 MPEG-4 meeting. Each object from the video is embedded with a unique watermark according its perceptual characteristics. As the object experiences translations and transformations over time, the watermark remains embedded with it. Objects defined in the video are collected into an object data base. As a result, the detection algorithm does not require information regarding the location (i.e., index) of the test frames in the video. The detection algorithm simply identifies the objects in the test frames. Once objects are identified, their watermarks may be retrieved from the data base and used to determine ownership.

Robustness to MPEG-1 coding at very high compression ratios (CR's) was tested on a 32-frame football video. The frame size of the sequence is 240 × 352. The MPEG quantization tables were set to the coarsest possible level to maximize compression. To simulate additional attacks on the watermark, colored noise was added to the test video prior to MPEG coding. The test video was tested 100 times, with a different colored noise sequence used during each run. The minimum, maximum, and mean frame-by-frame similarity values over the 100 runs are shown in Fig. 12. Even at very low coding quality, the similarity values are widely separated, allowing the existence of a watermark to be easily ascertained.

In a second approach [85], the authors employ a watermark that consists of fixed and varying components. A wavelet transform is applied along the temporal axis of the video to generate a multiresolution temporal representation of the video. The low-pass frames consist of the static components in the video scene. The high-pass frames capture the motion components and changing nature of the video sequence. The watermark is designed and embedded in each of these components. The watermarks embedded in the low-pass frames exist throughout the entire video scene due to wavelet localization properties. The watermarks embedded in the motion frames are highly localized in time and change rapidly from frame to frame. The resulting watermark is a composite of static and dynamic components.

For example, the plot in Fig. 13 shows the robustness of the technique to frame dropping and averaging. In the test, the odd index frames, i.e., $1, 3, \cdots$, were dropped from the test sequence. The missing frames were replaced with the average of the two neighboring frames, $F_{2n+1} = (F_{2n} + F_{2n+2})/2$. Colored noise of similar power to the watermark was added to the result. The resulting detection curves in the figure are shown to be widely separated. The error bars indicate the maximum and minimum similarity values over 100 runs with different colored noise sequences.

## VIII. Visible Marking of Images

Sophisticated, attractive, and robust visible watermarking methods for enhancing digital documents have been developed and patented [7] by Braudway *et al.* By robust,

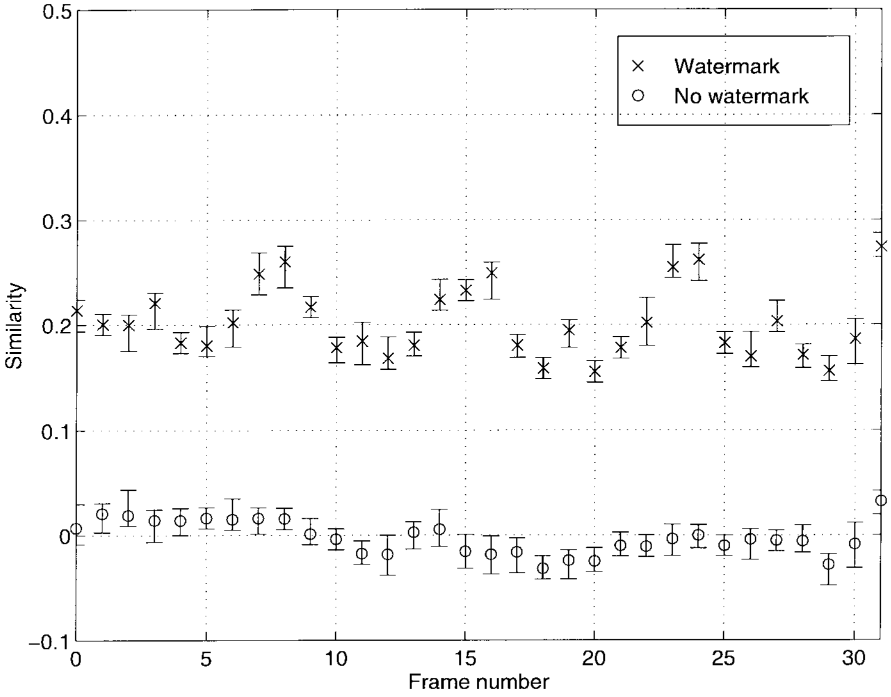


**Fig. 12.** Frame-by-frame robustness of a football video to MPEG coding at (0.46 Mbits/s, CR 44 : 1). The error bars around each similarity value indicate the maximum and minimum similarity values over the 100 runs.

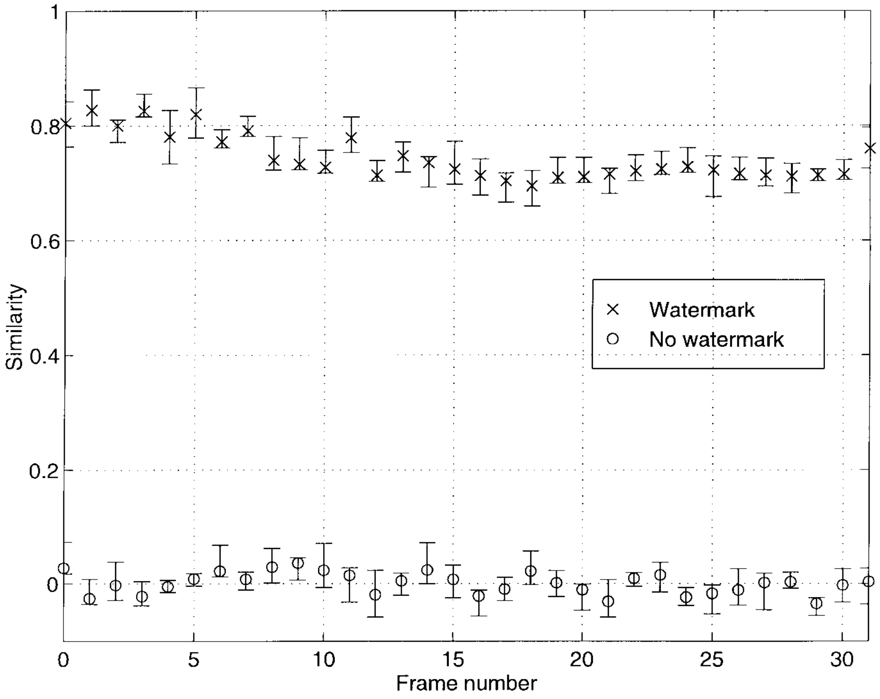


**Fig. 13.** Frame-by-frame robustness of a football video to frame dropping and averaging. The error bars around each similarity value indicate the maximum and minimum similarity values over the 100 runs.

we mean that the marks are difficult to remove without leaving a trace. The method of Braudway *et al.* for altering pixel values in a still image was used to mark digitized pages of manuscripts from the Vatican's archive and the British Library with a logo, in part for use in authenticating the images and in part for deterring any parties seeking to "purloin or misappropriate" the documents [8]; samples of marked images can be found at IBM.[7]

[7] See the IBM Digital Library home page, marked images from the Vatican Library Project http://www.software.ibm.com/is/dig-lib/vatican.html and marked images from British Library project http://www.rennard.demon. co.uk/tech/tewamk.htm.

To be attractive and effective when applied to digitized still-image data representing works with artistic merit, according to Braudway *et al.*, a visible watermark must be obvious to any person with normal or corrected vision (including the color blind), be flexible enough that it can be made as obtrusive or unobtrusive as desired, have bold features that (by themselves) form a recognizable image, allow all features of the unmarked image to appear in the marked image, and be very difficult, if not impossible, to remove.

The method designed by Braudway *et al.* to fulfill these criteria begins with the construction of a mask corresponding to the watermark. The mask determines which pixels in an image will remain unchanged and which will have their intensity altered. The mask is then resized, if necessary, to dimensions appropriate for the image size and marking purpose, and the location at which the watermark will be placed is chosen. Last, the intensity in the pixels specified by the mask is altered. The scientists used a mathematical model of the intensity in an image

$$\tilde{Y}_{m,n} = Y_{m,n} + C \times \Delta L^* \qquad (27)$$

where $Y_{m,n}$ and $\tilde{Y}_{m,n}$ represent the intensity of the $(m,n)$th pixel in the original and marked images, respectively, the constant $C$ is a function that reflects various properties of the specific image and watermark mask, and $L^*$ is the intensity (i.e., the amount of light received by the eye, regardless of color [81]). The appearance (or obtrusiveness) of the watermark is controlled by varying the intensity $L^*$. If the same value of $\Delta L^*$ were used to alter all the pixels that fall under the mask, then the watermark could be easily removed by a hostile party. To render robustness to the mark, randomness is introduced by using $2R_{m,n}\Delta L^*$ in place of $\Delta L^*$, where $R_{m,n} \in [0,1]$ is a discrete random variable that (if truly randomly distributed) satisfies

$$\lim_{M\to\infty} \lim_{N\to\infty} \frac{2}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} R_{m,n}\Delta L^* = \Delta L^*. \qquad (28)$$

A watermark needs to have bold features because the introduction of the random variable $R_{m,n}$, depending on its values, can make fine details of the mark less discernible. As an addendum to their method, Braudway *et al.* remark that additional robustness can be achieved by introducing small random variations in the size as well as in the horizontal and vertical placement of the watermark, as suggested by Pickerell and Child [67].

## References

[1] R. Anderson, Ed., “Information hiding,” in *Lecture Notes in Computer Science*. Tokyo, Japan: Springer, 1996.

[2] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, “Techniques for data hiding,” *IBM Syst. J.*, vol. 35, nos. 3 and 4, pp. 313–336, 1996.

[3] H. Berghel and L. O’Gorman, “Protecting ownership rights through digital watermarking,” *IEEE Computer Mag.*, pp. 101–103, July 1996.

[4] L. Boney, A. Tewfik, and K. Hamdy, “Digital watermarks for audio signals,” in *Proceedings of Multimedia’96*. Piscataway, NJ: IEEE Press, 1996, pp. 473–480.

[5] L. Boney, A. Tewfik, K. Hamdy, and M. Swanson, submitted for U.S. patent.

[6] J. Brassil, S. Low, N. Maxemchuk, and L. O’Gorman, “Electronic marking and identification techniques to discourage document copying,” *IEEE J. Select. Areas Commun.*, vol. 13, pp. 1495–1504, Oct. 1995.

[7] G. Braudway, K. Magerlein, and F. Mintzer, “Color correct digital watermarking of images,” U.S. Patent 5 530 759, June 25, 1996.

[8] ——, “Protecting publically-available images with a visible image watermark,” IBM Research Rep. TC-20336(89918), Jan. 15, 1996.

[9] O. Bruyndonckx, J. J. Quisquater, and B. Macq, “Spatial method for copyright labeling of digital images,” in *Proc. IEEE Nonlinear Signal Processing Workshop*, 1995, pp. 456–459.

[10] G. Caronni, “Assuring ownership rights for digital images,” in *Reliable IT Systems*, H. H. Brëggemann and W. Gerhardt-Häckl, Eds. Vieweg, Germany, 1995.

[11] M. Schneider and S.-F. Chang, “A content-based approach to image signature generation and authentication,” in *Proc. ICIP’96*, vol. III, pp. 227–230.

[12] “Facsimile coding schemes and coding control functions for group 4 facsimile apparatus for document transmission,” CCITT Recommendation T.6, 1984.

[13] M. Cooperman and S. Moskowitz, “Steganographic method and device,” U.S. Patent 5 613 004, Mar. 1997. (Available WWW: http://www.digital-watermark.com/patents.htm.)

[14] I. Cox, J. Kilian, T. Leighton, and T. Shamoon, “Secure spread spectrum watermarking for multimedia,” *IEEE Trans. Image Processing*, vol. 6, no. 12, pp. 1673–1687, 1997; see also *Proc. ICIP’96*, vol. III, pp. 243–246.

[15] S. Craver, N. Memon, B.-L. Yeo, and M. Yeung, “Can invisible watermarks resolve rightful ownership?” IBM Research Rep. RC20509, July 1996. (Available WWW: http://www.research.ibm.com:8080.) See also *Proc. SPIE Storage and Retrieval for Image and Video Databases V*, Feb. 1997, vol. 3022, pp. 310–321.

[16] ——, “Resolving rightful ownerships with invisible watermarking techniques: Limitations, attacks, and implications,” IBM Research Rep. RC 20755, Mar. 1997.

[17] C. Dautzenberg and F. Boland, “Watermarking images,” Dept. of Electrical Engineering, Trinity College, Dublin, Tech. Rep., 1994.

[18] P. Davern and M. Scott, “Fractal based image steganography,” in R. Anderson, Ed., *Lecture Notes in Computer Science*. Tokyo, Japan: Springer, 1996, pp. 279–294.

[19] W. Diffie and M. Hellman, “New directions in cryptography,” *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 644–654, 1976.

[20] P. Flikkema, “Spread-spectrum techniques for wireless communications,” *IEEE Trans. Signal Processing*, vol. 14, pp. 26–36, May 1997.

[21] J. Fridrich, “Methods for data hiding,” Center for Intelligent Systems & Dept. of Systems Science and Industrial Engineering, State Univ. of New York-Binghamton, preprint. (Available: http://ssie.binghamton.edu/jirif/.)

[22] B. Girod, “The information theoretical significance of spatial and temporal masking in video signals,” in *Proc. SPIE Human Vision, Visual Processing, and Digital Display*, 1989, vol. 1077, pp. 178–187.

[23] B. Girod and F. Hartung, submitted for U.S. patent.

[24] F. Goffin, J.-F. Delaigle, C. D. Vleeschouwer, B. Macq, and J.-J. Quisquater, “Low-cost perceptive digital picture watermarking method,” in *Proc. SPIE Storage and Retrieval for Image and Video Databases*, Jan. 1997, vol. 3022, pp. 264–277.

[25] S. Goldwasser and M. Bellare. (July 1996). Lecture notes on cryptography. [Online]. Available WWW: http://www-cse.ucsd.edu/users/mihir/papers/crypto-papers.html.

[26] B. Greenberg, “Method and apparatus for the processing of encoded data in conjunction with an audio broadcast,” U.S. Patent 5 379 345, Jan. 1995

[27] D. Gruhl. (1996). Examples of affine embedding. [Online]. Available WWW: http://nif.www.media.mit.edu/ DataHiding/affine/affine.html.

[28] D. Gruhl, A. Lu, and W. Bender, “Techniques for data hiding,” *IBM Syst. J.*, vol. 35, nos. 3 and 4, pp. 313–336, 1996.

[29] F. Hartung and B. Girod, “Digital watermarking of raw and compressed video,” in *Proc. SPIE Digital Compression Tech-*

*nologies and Systems for Video Communication*, Oct. 1996, vol. 2945, pp. 205–213.

[30] ——, “Copyright protection in video delivery networks by watermarking of pre-compressed video,” in *Multimedia Applications, Services and Techniques—ECMAST’97, Lecture Notes in Computer Science*, S. Fdida and M. Morganti, Eds. Tokyo, Japan: Springer, 1997, vol. 1242, pp. 423–436.

[31] ——, “Digital watermarking of MPEG-2 coded video in the bitstream domain,” in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing 1997*. Piscataway, NJ: IEEE Press, 1997, pp. 2621–2624.

[32] ——, “Digital watermarking of uncompressed and compressed video,” *Signal Processing*, to appear in 1997.

[33] ——, “Fast public-key watermarking of compressed video,” in *Proceedings of the IEEE International Conference on Image Processing 1997*. Piscataway, NJ: 1997., vol. I, pp. 528–531.

[34] Herodotus, *The Histories (trans. A. de Selincourt)*. Middlesex, England: Penguin, 1972.

[35] Homer, *The Iliad (trans. R. Fagels)*. Middlesex, England: Penguin, 1990.

[36] C.-T. Hsu and J.-L. Wu, “Hidden signatures in images,” in *Proceedings of the IEEE International Conference on Image Processing 1996*. Piscataway, NJ: IEEE Press, 1996, pp. 223–226.

[37] *Proceedings of the International Conference on Acoustics, Speech and Signal Processing 1997*. Piscataway, NJ: IEEE Press, 1997.

[38] *Proceedings of the IEEE International Conference on Image Processing 1996*. Piscataway, NJ: IEEE Press, 1996.

[39] *Proceedings of the IEEE International Conference on Image Processing 1997*. Piscataway, NJ: 1997.

[40] “Information technology—Coding of moving pictures and associated audio for digital storage up to about 1.5 Mbits/s,” ISO/IEC IS 11172, 1993.

[41] N. Jayant, J. Johnston, and R. Safranek, “Signal compression based on models of human perception,” *Proc. IEEE*, vol. 81, pp. 1385–1422, Oct. 1993.

[42] J. Johnston and K. Brandenburg, “Wideband coding—Perceptual considerations for speech and music,” in *Advances in Speech Signal Processing*, S. Furui and M. Sondhi, Eds. New York: Marcel Dekker, 1992.

[43] D. Kahn, *The Codebreakers*. New York: MacMillan, 1967.

[44] D. Knuth, *The Art of Computer Programming*, vol. 2, 2nd ed. Menlo Park, CA: Addison-Wesley, 1981.

[45] M. Kuhn. (1997). StirMark, [Online]. Available WWW: http://www.cl.cam.ac.uk/∼fapp2/ watermarking/image_watermarking/stirmark.

[46] G. Langelaar, J. van der Lubbe, and J. Biemond. (1996). Copy protection for multimedia based on labeling techniques. Available WWW: http://www-it.et.tudelft.nl/pda/smash/public/benelux_cr. html.

[47] C. Lee, K. Moallemi, and J. Hinderling, “Post-compression hidden data transport,” U.S. Patent 5 687 191, 1997.

[48] G. Legge and J. Foley, “Contrast masking in human vision,” *J. Opt. Soc. Amer.*, vol. 70, no. 12, pp. 1458–1471, Dec. 1990.

[49] S. Low, N. Maxemchuk, J. Brassil, and L. O’Gorman, “Document marking and identification using both line and word shifting,” in *Proc. Infocom’95*, Boston, MA, Apr. 1995. (Available WWW: http://www.research.att.com:80/ lateinfo/projects/ecom.html.)

[50] Kinoshita, Inaba, and Kasahara, “Digital watermarking of video,” in *Proc. 1997 Symp. Cryptography and Information Security*, Jan. 1997, SCIS97-31F (in Japanese).

[51] E. Koch and J. Zhao, “Embedding robust labels into images for copyright protection,” in *Proc. Int. Congress Intellectual Property Rights for Specialized Information, Knowledge, and New Technologies*, Vienna, Austria, Aug. 1995, pp. 242–251.

[52] M. Kutter, F. Jordan, and F. Bossen, “Digital signature of color images using amplitude modulation,” in *Proc. SPIE-EI97*, 1997, pp. 518–526.

[53] M. Luby, *Pseudorandomness and Cryptographic Applications*. Princeton, NJ: Princeton Univ. Press, 1996.

[54] K. Matsui and K. Tanaka, *Gazo Shinso Ango*. Morikita, 1993 (in Japanese).

[55] ——, “Video-steganography: How to embed a signature in a picture,” in *Proc. IMA Intellectual Property*, Jan. 1994, vol. 1, no. 1, pp. 187–206.

[56] N. Maxemchuk, “Electronic document distribution,” *AT&T Tech. J.*, pp. 73–80, Sept. 1994.

[57] D. Moses, “Simultaneous transmission of data and audio signals by means of perceptual coding,” U.S. Patent 5 473 631, 1995.

[58] W. Mowry, Jr., M. McElligott, V. Tkalenko, J. Baran, and C Ingalls, “Protected document bearing watermark and method of making,” U.S. Patent 4 210 346, July 1, 1980.

[59] *Proceedings of Multimedia’96*. Piscataway, NJ: IEEE Press, 1996.

[60] National Institute of Standards and Technology (NIST), *Secure Hash Standard*, NIST FIPS Pub. 180-1, Apr. 1995.

[61] P. Noll, “Wideband speech and audio coding,” *IEEE Commun. Mag.*, pp. 34–44, Nov. 1993.

[62] R. Ohbuchi, H. Masuda, and M. Aono, “Embedding data in three-dimensional models,” in *Proc. Eur. Workshop Interactive Distributed Multimedia Systems and Telecommunication Services, Darmstadt, Germany, Lecture Notes in Computer Science*, no. 1309. Tokyo, Japan: Springer, 1997.

[63] ——, “Watermarking three-dimensional models,” in *Proc. 5th ACM Int. Multimedia Conf.*, Seattle, WA, Nov. 9–13, 1997, pp. 261–272.

[64] J. Ohnishi and K. Matsui, “Embedding a seal into a picture under orthogonal wavelet transform,” in *Proceedings of Multimedia’96*. Piscataway, NJ: IEEE Press, 1996., pp. 514–521.

[65] J. O’Ruanaidh, C. Dautzenberg, and F. Boland, “Watermarking digital images for copyright protection,” *Proc. Inst. Elect. Eng. Vis. Image Signal Processing*, Aug. 1996, vol. 143, no. 4, pp. 250–256.

[66] ——, “Phase watermarking of digital images,” in *Proceedings of the IEEE International Conference on Image Processing 1996*. Piscataway, NJ: IEEE Press, 1996. pp. 239–242.

[67] J. Pickerell and A. Child, “Marketing photography in the digital environment,” DiSC, Rockville, MD, 1994.

[68] R. Pickholz, D. Schilling, and L. Milstein, “Theory of spread spectrum communications,” *IEEE Trans. Commun.*, vol. COM-30, pp. 855–884, May 1982.

[69] I. Pitas, “A method for signature casting on digital images,” in *Proceedings of the IEEE International Conference on Image Processing 1996*. Piscataway, NJ: IEEE Press, 1996., vol. III, pp. 215–218.

[70] I. Pitas and T. Kaskalis, “Applying signatures on digital images,” in *Proc. 1995 IEEE Nonlinear Signal Processing Workshop*, 1995, pp. 460–463.

[71] C. Podilchuk and W. Zeng, “Digital image watermarking using visual models,” Multimedia Communication Lab, Lucent Technologies, Bell Labs, Tech. Memo, Sept. 1996. See also *Proc. SPIE/IS&T Electronic Imaging’97: Human Vision and Electronic Imaging*, Feb. 1997, vol. 3022, pp. 310–321.

[72] ——, “Perceptual watermarking of still images,” in *Proc. IEEE Workshop Multimedia Signal Processing*, June 1997, pp. 363–368.

[73] R. Preuss, S. Roukos, A. Huggins, H. Gish, M. Bergamo, and P. Peterson, “Embedded signalling,” U.S. Patent 5 319 735, 1994.

[74] J. Puate and F. Jordan. (1996). Using fractal compression scheme to embed a digital signature into an image. [Online]. Available: http://lswww.epfl.ch/˜jordan/watermarking.html.

[75] J. Ragusa, V. Badari, and J. Machuca. (1997). An adaptive spread spectrum approach. [Online]. Available WWW: http://www.csuglab.cornell.edu/Info/People/vbadari/cs631/wmrkproj/proposal.html.

[76] R. Rivest, “Cryptography,” in *Handbook of Theoretical Computer Science*, vol. 1, J. van Leeuwen, Ed. Cambridge, MA: MIT Press, 1990, ch. 13, pp. 717–755.

[77] ——, “The MD4 message digest algorithm,” in *Advances in Cryptology, CRYPTO’92*. Tokyo, Japan: Springer, 1991, pp. 303–311.

[78] K. Rosen, *Elementary Number Theory and Its Applications*, 3rd ed. Tokyo, Japan: Addison-Wesley, 1992.

[79] J. Russ, *The Image Processing Handbook*, 2nd ed. Tokyo, Japan: CRC Press, 1995.

[80] M. Sanford, J. Bradley, and T. Handel, “The data embedding method,” Los Alamos National Laboratory Rep. 9LA-95-2246UR, Sept. 25, 1995. (Available WWW: http://www.lanl.gov/users/u078743/embed1.htm.)

[81] M. Sid-Ahmed, *Image Processing*. Tokyo, Japan: McGraw-Hill, 1995.

[82] J. Smith and B. Comiskey, “Modulation and information hiding in images,” in R. Anderson, Ed., “Information hiding,” in *Lecture Notes in Computer Science*. Tokyo, Japan: Springer, 1996, pp. 207–226.

[83] H. Stone, “Analysis of attacks on image watermarks with randomized coefficients,” NEC Research Institute, Princeton, NJ, Tech. Rep., May 17, 1996. (Available WWW: http://www.neci.nj.nec.com.)

[84] M. Swanson, B. Zhu, and A. Tewfik, “Data hiding for video in video,” in *Proceedings of the IEEE International Conference on Image Processing 1997*. Piscataway, NJ: 1997, vol. II, pp. 676–679.

[85] M. Swanson, B. Zhu, and A. Tewfik, “Multiresolution video watermarking using perceptual models and scene segmentation,” *IEEE J. Select. Areas Commun.*, to be published. See also *Proceedings of the IEEE International Conference on Image Processing 1997*. Piscataway, NJ: 1997, vol. II, pp. 558–561.

[86] M. Swanson, B. Zhu, and A. Tewfik, “Object-based transparent video watermarking,” in *Proc. 1997 IEEE Multimedia Signal Processing Workshop*, 1997, pp. 369–374.

[87] M. Swanson, B. Zhu, A. Tewfik, and L. Boney, “Robust audio watermarking using perceptual masking,” *Signal Process.*, to be published.

[88] M. Swanson, B. Zhu, and A. Tewfik, “Transparent robust image watermarking,” in *Proceedings of the IEEE International Conference on Image Processing 1996*. Piscataway, NJ: IEEE Press, 1996, vol. III, pp. 211–214.

[89] J. Tilki and A. Beex, “Encoding a hidden digital signature onto an audio signal using psychoacoustic masking,” in *Proc. 1996 7th Int. Conf. Sig. Proc. Appls. Tech.*, 1996, pp. 476–480.

[90] R. van Schyndel, A. Tirkel, and C. Osborne, “A digital watermark,” in *Proceedings of ICASSP*. Piscataway, NJ: IEEE Press, 1994, vol. II, pp. 86–90.

[91] H. van Trees, *Detection, Estimation, and Modulation Theory*, vol. I. New York: Wiley, 1968.

[92] A. Viterbi, *CDMA Principles of Spread Spectrum Communication*. Tokyo, Japan: Addison-Wesley, 1995.

[93] P. Vogel, “System for altering elements of a text file to mark documents,” U.S. Patent 5 388 194, Feb. 7, 1995.

[94] G. Voyatzis and I. Pitas, “Applications of toral automorphisms in image watermarking,” in *Proceedings of the IEEE International Conference on Image Processing 1996*. Piscataway, NJ: IEEE Press, 1996, vol. II, pp. 237–240.

[95] S. Walton, “Image authentication for a slippery new age,” *Dr. Dobb’s J.*, pp. 18–26 and 82–87, Apr. 1995.

[96] R. Wolfgang and E. Delp, “A watermark for digital images,” in *Proceedings of the IEEE International Conference on Image Processing 1996*. Piscataway, NJ: IEEE Press, 1996, pp. 219–222.

[97] ——, “A watermarking technique for digital imagery: Further studies,” in *Proc. Int. Conf. Imaging Science, Systems and Technology*, Las Vegas, NV, June 30–July 3, 1997.

[98] J. Zhao and Fraunhofer Inst. for Computer Graphics. (1996). [Online]. Available WWW: http://www.igd.fhg.de/~zhao/zhao.html.

[99] B. Zhu, A. Tewfik, and O. Gerek, “Low bit rate near-transparent image coding,” in *Proc. SPIE Int. Conf. Wavelet Appls. for Dual Use*, 1995, vol. 2491, pp. 173–184.

**Mitchell D. Swanson** (Member, IEEE) received the B.S. (*summa cum laude*), M.S., and Ph.D. degrees in electrical engineering from the University of Minnesota, Minneapolis, in 1992, 1995, and 1997, respectively.

He was with Honeywell, Inc., and Medtronic, Inc., Minneapolis. He currently is with Cognicity, Inc., Minneapolis, and is a Visiting Assistant Professor with the Department of Electrical Engineering at the University of Minnesota. His research interests include multiscale signal processing, image and video coding for interactive retrieval, data hiding, and digital watermarking.

**Mei Kobayashi** received the A.B. degree in chemistry from Princeton University, Princeton, NJ, in 1981 and the M.A. and Ph.D. degrees in pure and applied mathematics from the University of California at Berkeley in 1984 and 1988, respectively.

During her undergraduate and graduate years, she was an Intern in the Chemistry and Physics Divisions at Lawrence Berkeley Laboratories. She has been with the IBM Tokyo Research Laboratory, Japan, since April 1988. From April 1996 to March 1999, she will be a Visiting Associate Professor in the Graduate School of Mathematical Sciences at the University of Tokyo, Tokyo, Japan.

**Ahmed H. Tewfik** (Fellow, IEEE) was born in Cairo, Egypt, on October 21, 1960. He received the B.Sc. degree from Cairo University in 1982 and the M.Sc., E.E., and Sc.D. degrees from the Massachusetts Institute of Technology, Cambridge, in 1984, 1985, and 1987, respectively.

He was with Alphatech, Inc., Burlington, MA, in 1987. He currently is the E. F. Johnson Professor of Electronic Communications with the Department of Electrical Engineering at the University of Minnesota, Minneapolis. He was a Consultant to MTS Systems, Inc., Eden Prairie, MN, and is a regular Consultant to Rosemount, Inc., Eden Prairie. His current research interests are in signal processing for multimedia (in particular watermarking, data hiding, and content-based retrieval), low-power multimedia communications, adaptive search and data-acquisition strategies for World Wide Web applications, radar and dental/medical imaging, monitoring of machinery using acoustic emissions, and industrial measurements.

Dr. Tewfik is a Distinguished Lecturer of the IEEE Signal Processing Society for July 1997–July 1999. He was a Principal Lecturer at the 1995 IEEE EMBS summer school. He received a Taylor Faculty Development Award from the Taylor Foundation in 1992 and a National Science Foundation Research Initiation Award in 1990. He gave a plenary lecture at the 1994 IEEE International Conference on Acoustics, Speech, and Signal Processing and an invited tutorial on wavelets at the 1994 IEEE Workshop on Time-Frequency and Time-Scale Analysis. He became the first Editor-in-Chief of IEEE SIGNAL PROCESSING LETTERS in 1993. He is a past Associate Editor of IEEE TRANSACTIONS ON SIGNAL PROCESSING and was a Guest Editor of two special issues on wavelets and their applications.