

Digital Watermarking of Low Bit-Rate Advanced Simple Profile MPEG-4 Compressed Video

Adnan M. Alattar, *Member, IEEE*, Eugene T. Lin, *Student Member, IEEE*, and Mehmet Utku Celik, *Student Member, IEEE*

Abstract—A novel MPEG-4 compressed domain video watermarking method is proposed and its performance is studied at video bit rates ranging from 128 to 768 kb/s. The spatial spread-spectrum watermark is embedded directly to compressed MPEG-4 bitstreams by modifying DCT coefficients. A synchronization template combats geometric attacks, such as cropping, scaling, and rotation. The method also features a gain control algorithm that adjusts the embedding strength of the watermark depending on local image characteristics, increasing watermark robustness or, equivalently, reducing the watermark's impact on visual quality. A drift compensator prevents the accumulation of watermark distortion and reduces watermark self-interference due to temporal prediction in inter-coded frames and AC/DC prediction in intra-coded frames. A bit-rate controller maintains the bit rate of the watermarked video within an acceptable limit. The watermark was evaluated and found to be robust against a variety of attacks, including transcoding, scaling, rotation, and noise reduction.

Index Terms—MPEG-4, spread spectrum, synchronization template, video watermarking.

I. INTRODUCTION

THE INTERNET and other digital networks offer free and wide distribution of high-fidelity duplicates of digital media, which is a boon for authorized content distribution. However, these networks are also an avenue for major economic loss arising from illicit distribution and piracy. An effective digital-rights-management (DRM) system lets content providers track, monitor, and enforce usage rights in both digital and analog form. A DRM system can also link users to content providers and may promote sales.

Encryption and watermarking [1]–[4] are two oft-mentioned techniques proposed for use in DRM systems. Although encryption plays an important role in DRM and video streaming, it can only protect the digital content during transmission from the content provider to the authorized user. Once the content has been decrypted, encryption no longer provides any protection. In contrast, a watermark persists within the decrypted video stream and can be used to control access to the video. A DRM-compliant device can read the embedded watermark and control or prevent video playback and duplication according to

the information contained in the watermark. A watermark even persists in the video when it has been converted from digital to analog form. Video watermarking may also be used for tracking or tracing video content and broadcast monitoring, as well as for linking the video to its provider and facilitating value-added services that benefit both the providers and the consumers.

The classical approach to watermark a compressed video stream is to decompress the video, use a spatial-domain or transform-domain watermarking technique, and then recompress the watermarked video. There are three major disadvantages to using this classical approach. First, the watermark embedder has no knowledge of how the video will be recompressed and cannot make informed decisions based on the compression parameters. This approach treats the video compression process as a removal attack and requires the watermark to be inserted with excessive strength, which can adversely impact watermark perceptibility. Moreover, a second compression step is likely to add additional compression noise, degrading the video quality further. Finally, fully decompressing and re-compressing the video stream can be computationally expensive.

A faster and more flexible approach to watermarking compressed video is that of *compressed-domain watermarking*. In compressed-domain watermarking, the original compressed video is partially decoded to expose the syntactic elements of the compressed bitstream for watermarking (such as encoded DCT coefficients.) Then, the partially decoded bitstream is modified to insert the watermark and lastly, reassembled to form the compressed watermarked video. The watermark insertion process ensures that all modifications to the compressed bitstream will produce a syntactically valid bitstream that can be decoded by a standard decoder. In contrast with the classical approach, the watermark embedder has access to information contained in the compressed bitstream, such as prediction and quantization parameters, and can adjust the watermark embedding accordingly to improve robustness, capacity, and visual quality. In addition, this approach lets a watermark be embedded without resorting to the computationally expensive motion estimation process during recompression. Note that the corresponding computational gain is highly dependent on the particular implementation of the motion estimation process. Hartung [5], [6] describes techniques to embed a spread-spectrum watermark into MPEG-2 [7] compressed video (using compressed-domain embedding), as well as into uncompressed video (classical approach). For compressed-domain watermark embedding, Hartung's technique partially decodes the MPEG-2 video to obtain the DCT coefficients of each frame and inserts the watermark by modifying those

Manuscript received December 16, 2002; revised April 20, 2003.

A. M. Alattar is with Digimarc Corporation, Tualatin, OR 97062 USA (e-mail: aalattar@digimarc.com).

E. T. Lin is with the School of Computer and Electrical Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: linet@ecn.purdue.edu).

M. U. Celik is with the Electrical and Computer Engineering Department, University of Rochester, Rochester, NY 14627 USA (e-mail: celik@ece.rochester.edu).

Digital Object Identifier 10.1109/TCSVT.2003.815958

DCT coefficients. The technique includes a method for drift compensation. Data rate control is performed by watermarking only nonzero DCT coefficients, and only if the data rate will increase as a result of watermarking. Hartung evaluated the technique for compressed videos with rates between 4 and 12 Mb/s, which are more suitable for DVD and digital TV broadcast than for low data-rate video (< 1 Mb/s).

In this paper, we present a new compressed-domain watermarking technique for MPEG-4 [8] video streams. Our approach is similar to Hartung's in that we perform partial decoding and embed the watermark into the DCT coefficients of a compressed video stream. However, our approach has several new and enhanced features over Hartung's. Our drift compensation method supports the prediction modes in MPEG-4, including spatial (intra-DC and intra-AC) prediction. Our watermark detection is performed in the spatial domain with templates to establish and maintain detector synchronization. In addition, our technique introduces new methods for adapting the gain of the watermark based on the characteristics of the original video and for controlling the data rate of the watermarked video. Experimental results indicate that our technique is robust against a variety of attacks including filtering, scaling, rotation, and transcoding.

An overview of MPEG-4 and video watermarking is presented in Section II, our watermarking technique is described in Section III, followed by results in Section IV. Conclusions are presented in Section V.

II. BACKGROUND

A. MPEG-4

MPEG-4 is an object-based standard for coding multimedia at low bit rates (< 1 Mb/s) [8]. MPEG-4 encodes the visual information as objects, which include natural video, synthetic video (mesh and face coding of wire frame), and still texture. In addition, MPEG-4 encodes a description of the scene for proper rendering of all objects. At the decoding end, the scene description and the individual media objects are decoded, synchronized, and composed for presentation. This paper is limited to natural video; it does not address synthetic video (3-D objects) nor still texture.

A natural video object (VO) in MPEG-4 may correspond to the entire scene or a physical object in the scene. A physical object is expected to have a semantic meaning such as car, tree, or person. A video object plane (VOP) is a temporal instance of a VO, and a displayed frame is the overlap of the same instance VOPs of all video objects in the sequence. A frame is only composed during the display process using information provided by the encoder or the user. This information indicates where and when VOPs of a VO are displayed. Video objects may have arbitrary shapes. The shape information is encoded using a context-switched arithmetic encoder that is provided along with the texture information.

The texture information is encoded using a hybrid motion-compensated DCT compression algorithm similar to that used in MPEG-1 and MPEG-2. This algorithm uses motion compensation to reduce inter-frame redundancy and the DCT to compact the energy in every 8×8 block of the image into a few

coefficients. The algorithm then adaptively quantizes the DCT coefficients to achieve the desired bit rate. Huffman codes are used by the algorithm to encode the quantized DCT coefficients, the motion vectors, and most control parameters to reduce the statistical redundancies in the data. All coded information is assembled into an elementary bitstream that represents a single video object. MPEG-4 has enhanced coding efficiency that can be attributed partially to sophisticated DC coefficient, AC coefficient, and motion vector prediction algorithms, as well as to overlapped block motion compensation.

To enable using MPEG-4 with many applications, MPEG-4 includes a variety of encoding tools. MPEG-4 allows the encoding of interlaced as well as progressive video. It also allows temporal and spatial scalability. Moreover, it allows sprite encoding. However, not all of these tools are needed for a particular application. Hence, to simplify the design of the decoders, MPEG-4 defines a set of profiles and a set of levels within each profile. Each profile was designed with one class of applications in mind. Simple, Advanced Simple, Core, Main, and Simple Scalable are some of the profiles for natural video.

Video compression field tests of natural video at rates below 1 Mb/s have indicated consistently better performance for MPEG-4 than for MPEG-1 and MPEG-2. Increased compression efficiency and flexibility of the standard prompted Internet Streaming Media Alliance to promote Advanced Simple Profile (ASP) of MPEG-4 for broadband Internet multimedia streaming. ASP supports all capabilities of MPEG-4 Simple Profile in addition to B-VOPs, quarter-pel motion compensation, extra quantization tables, and global motion compensation. However, ASP does not support arbitrary-shaped objects, scalability, interlaced video, nor sprites.

The discussion of this paper will be limited to watermarking of natural video sequences that are compressed according to ASP. However, the techniques and methodology employed in this paper can be easily extended to the Core, Main, and Simple Scalable profiles, often with only minor modifications needed. Watermarking 3-D objects will not be considered in this paper.

B. Video Watermarking

Digital video presents many challenges for watermarking. Foremost, many digital video applications employ lossy compression techniques such as MPEG-1 [9], MPEG-2 [7] and MPEG-4 [8]. To achieve an efficient representation of the video, compression techniques remove spatial, temporal, and perceptual redundancy from the video. Unfortunately from a robust watermarking perspective, lossy compression is considered a form of attack, as this compression may severely damage a watermark by removing parts of watermark signal. The computational cost of watermark embedding and detection is another challenge in video watermarking. For example, this cost is especially relevant in real-time watermarking of live video streams [10] or just-in-time watermarking for video-on-demand applications.

Furthermore, compressed domain watermarking introduces problems that do not apply in the classical approach. Watermark embedding must be coupled tightly with a specific compression method. This coupling not only restricts the portability of the watermarking algorithm, but also imposes limitations set

forth by the bitstream syntax and coding algorithm. The second problem is that of drift when the video is modified during watermark insertion. Drift occurs when (spatial or temporal) prediction is used and a predictor is modified without adjusting the residual to compensate for the new predictor. A compressed-domain watermarking technique must compensate for drift during watermark insertion to prevent drift from spreading and accumulating, leading to visible artifacts in the decoded video. Another challenge is adjusting the local strength of the watermark according to the properties of the human visual system without accessing the fully decompressed video. Lastly, the data rate of the compressed stream may substantially increase due to watermarking. Hence, the data rate must be controlled to remain within acceptable limits.

In blind watermarking techniques, where the unwatermarked original is not available at the decoder, the detector must synchronize [4], [11] with the spatial and temporal coordinates of the watermark signal for reliable detection. De-synchronization may occur as a result of a benign operation, such as changing the format to match a particular screen size (e.g., PanScan and letterbox) or as a result of a malicious attack to render the watermark undetectable. The most fundamental method for establishing synchronization between the detector and the watermark is a search over the space of all possible transformations (translations, rotations, scales, warping) until synchronization is found or the detector decides there is no watermark present [5]. However, this is not practical for video applications where the search space for transformations is much too large. A practical means for establishing and maintaining synchronization in video is the embedding of a template, which can be examined by the watermark detector to determine the orientation and scale of the watermark. Efficient synchronization is achieved by selecting the embedding domain [12] or by appropriate design of the watermark [11], [13], [14].

Compressed domain watermarking has been examined in the literature. In addition to Hartung's method [5], Langelaar and Lagendijk [15] and Setyawan and Lagendijk [16] describe a compressed domain watermarking technique called differential energy watermark (DEW) in which the watermark is inserted into DCT coefficients. The video is partitioned into groups of blocks, each of which is further divided into two sets of equal size, as determined by the watermark embedding key. By comparing the energy of selected DCT coefficients within the two sets, a single payload bit is expressed. If necessary, the energy of the sets of blocks is adjusted (by zeroing DCT coefficients) to express the desired payload bit. The technique is not very robust against transcoding, particularly if the GOP structure is changed. Also, the DEW watermark was examined for high-rate video (> 4 Mb/s), which has many more nonzero DCT coefficients available for watermark embedding than low-rate video.

Several researchers evaluated watermarking in the context of MPEG-4 compression. The authors of [17]–[19] investigated the watermarking of individual video objects in the spatial uncompressed domain. Nicholson [20] evaluated watermark robustness and video quality after the video was watermarked and compressed by MPEG-4 standard at bit rates ranging from 0.250 to 8 Mb/s. However, none of these techniques address direct watermarking of MPEG-4 compressed bitstreams. Hartung *et al.*

proposed a technique for watermarking MPEG-4 facial animation parameter data sets [21].

III. PROPOSED METHOD

In the proposed method, a watermark signal is inserted directly into the MPEG-4 compressed bitstream while detection is performed using the uncompressed video. This method allows detection if video has been manipulated or its format changed, without writing a detector to interpret new formats.

The elementary watermark signal is designed in the uncompressed pixel domain and is consecutively inserted directly into the MPEG-4 bitstream. Using a spatial-domain elementary watermark signal simplifies the correspondence between the compressed domain embedding and pixel domain detection processes. The elementary watermark signal consists of a spread-spectrum message signal and a synchronization template. Section III-A outlines the design of the spread-spectrum message signal for coping with host signal interference and subsequent processing noise and two synchronization templates for coping with possible geometrical manipulations. Section III-B addresses the process in which the elementary watermark signal is inserted into the MPEG-4 bitstream. Hartung's approach for MPEG-2 domain watermarking, which embeds the watermark signal by modifying DCT-coefficients, is extended to MPEG-4 and its extended features. In addition, a novel gain control algorithm designed for compressed domain implementation, a drift compensator that prevents accumulation of watermark distortion and self-interference, and a novel bit-rate control mechanism are presented.

A. Elementary Spread-Spectrum Watermark

Our elementary watermark is a spread-spectrum signal in spatial domain and covers the entire video object. In direct-sequence spread-spectrum communications, the message signal is modulated with a pseudo-noise pattern and detection is performed using a correlation detector [22, pp. 578–611]. Spread-spectrum communication techniques provide reliable data transmission even in very low signal-to-noise ratio (SNR) conditions. The watermark signal is often limited to a small value to ensure the imperceptibility and subject to interference from the host signal and additional noise arising from subsequent processing. As a result, spread-spectrum techniques are frequently used in watermarking applications and their use is studied extensively in the literature [23], [24].

Despite its robustness against additive noise, a spread-spectrum watermark is vulnerable to synchronization error, which occurs when the watermarked signal undergoes geometric manipulations such as scaling, cropping, and rotation. Before proceeding to the encoding of the message payload using spread-spectrum techniques, we outline template-based mechanisms that combat loss of synchronization.

1) *Synchronization Templates*: A template is any pattern or structure in the embedded watermark that can be exploited to recover synchronization at the decoder and is not limited to the addition of auxiliary signals as often referred to in the literature [25], [26]. Here, a pair of templates is imposed on the spread-spectrum signal to combat synchronization loss. In particular,

the synchronization templates are used to determine the change in rotation and scale after watermark embedding. Once known, these modifications are reversed prior to detection of the spread-spectrum message.

The first template is implicit in the signal design and restricts the watermark signal to have a regular (periodic) structure. In particular, the watermark $w(x, y)$ is constructed by repeating an elementary watermark tile $\hat{w}(x, y)$ (of size $N \times M$) in a nonoverlapping fashion. This tiled structure of the watermark can be detected easily by autocorrelation [22, pp. 271–273]. If the watermark tile is designed appropriately, a peak occurs at the center of each tile. When a pseudorandom noise pattern with a white power spectrum is used as the watermark tile, periodic impulses are observed in the autocorrelation domain. A colored noise pattern is often used in practical applications, at the expense of sharper peaks.

If a linear transformation A is applied to a watermarked VOP, the autocorrelation coefficients $h(x, y)$, thus the peaks, move to new locations (x', y') according to

$$[x' \ y']^T = A[x \ y]^T. \quad (1)$$

A similar approach has been described by Kalker and *et al.* [27] and Delanny and Macq [28].

The second synchronization template forces $w(x, y)$ to contain a constellation of peaks in the frequency domain. This requirement can be met by constructing $\hat{w}(x, y)$ as a combination of an explicit synchronization signal, $g(x, y)$, and a message-bearing signal $m(x, y)$. A similar approach has been described by O'Ruanaidh and Pun [29]. In the frequency domain, $g(x, y)$ is composed of peaks in the mid-frequency band, each peak occupying one frequency coefficient and having unity magnitude and pseudorandom phase. The random phase makes the signal look somewhat random in the spatial domain. Since the magnitude of the fast Fourier transform (FFT) is shift invariant and a linear transformation applied to the image has a well-understood effect on the frequency representation of the image, these peaks can be detected in the frequency domain and used to combat geometrical distortions. Specifically, a linear transformation A applied to the image f will cause its FFT coefficient $F(u, v)$ to move to a new location (u', v') , such that

$$[u' \ v']^T = (A^T)^{-1}[u \ v]^T. \quad (2)$$

Note that the magnitude of $F(u, v)$ will be scaled by $|A|^{-1/2}$.

If A represents a uniform scaling by factor S and a counter-clockwise rotation by angle θ , then

$$A = \begin{bmatrix} S \cos \theta & -S \sin \theta \\ S \sin \theta & S \cos \theta \end{bmatrix}. \quad (3)$$

The unknown scaling and rotation parameters can be obtained using either or both of the synchronization templates. A log-polar transform of the coordinates is used to convert the scale and rotation into linear shifts in the horizontal and vertical directions. For synchronization, using the first template (autocorrelation) the origin of the log-polar mapping is chosen as the

largest peak (image center). Under the log-polar mapping, the coordinate transformation of (1) becomes

$$\begin{bmatrix} \log \rho' \\ \alpha' \end{bmatrix} = \begin{bmatrix} \log \rho \\ \alpha \end{bmatrix} + \begin{bmatrix} \log S \\ \theta \end{bmatrix}. \quad (4)$$

For the second template (Fourier coefficients), the mapping will have the same form as (4) with a different scale term ($1/S$ or negative shift in scale direction). Given that the watermark templates are known, the linear shifts in log-polar domain can be detected using a phase-only match filter (POM).

2) *Message Signal Formation*: The message-bearing signal is constructed using the tiling pattern enforced by the synchronization template. In particular, a message signal tile, $m(x, y)$, of size $N \times M$ is formed to carry the required payload. A 31-bit payload was used for watermarking each MPEG-4 video object. Error correction and detection bits were added to the message to protect it from channel errors caused by the host image or distortion noise added by normal processing or an intentional attacker.

To reduce visibility and the effect of the host image on the watermark, spread-spectrum modulation is used with the message bits. First, the values 0,1 are mapped to -1 and 1 , respectively. Then, each bit is multiplied by a different pseudorandom code of length K producing a spread vector of size K . Finally, an $N \times M$ tile is constructed using all the resulting spread vectors by scattering them over the tile, such that each location of the tile is occupied by a unique bit. This permutation has a similar effect to whitening the image signal before adding the watermark, which improves the performance of the correlator used by the watermark detector. This tile comprises the message signal $m(x, y)$.

The watermark tile signal, $\hat{w}(x, y)$, was composed by adding the message signal $m(x, y)$ to the spatial representation of the synchronization signal $g(x, y)$ as

$$\hat{w}(x, y) = am(x, y) + bg(x, y) \quad (5)$$

where a and b are predetermined constants that control relative power between the message and the synchronization signals. These coefficients are adjusted according to the expected distortions in the operating environment and underlying host signal characteristics. For instance, if robustness against additive noise is necessary while geometric manipulations are less probable, the power of the message signal is increased in the expense of synchronization signal power. The signal $g(x, y)$ provides additional synchronization capability especially at all low bit rates where a large part of the watermark signal is expected to be lost due to the coarse quantization of the watermarked DCT coefficients. Fig. 1 illustrates the steps for creating the watermark including the two synchronization templates.

B. Watermarking MPEG-4 Compressed Domain Video

This section describes embedding the watermark directly to the bitstream generated in accordance with the ASP of the MPEG-4 standard.

1) *Watermark Embedding*: The watermark embedder mimics the system decoder model described by the MPEG-4 standard [8]. The Delivery layer extracts access units (SL

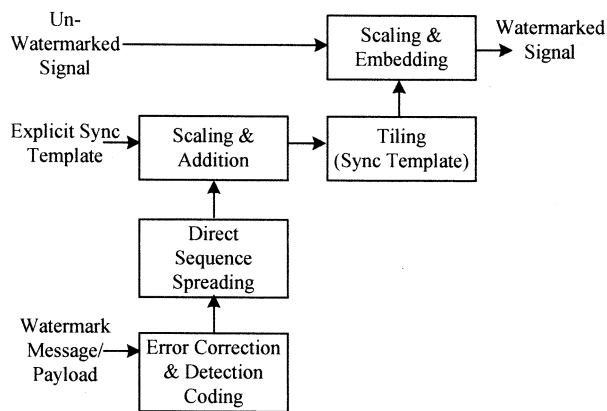


Fig. 1. Formation of elementary watermark signal. A message signal and an explicit synchronization template are combined and tiled.

packet) and their associated framing information from the network or storage device and passes them to the Sync Layer. The Sync layer extracts the payloads from the SL packets and uses the stream map information to identify and assemble the associated elementary bitstreams. Finally, the elementary bitstreams are parsed and watermarked according to the scene description information. The Sync layer re-packetizes the elementary bitstreams into access units and delivers them to the Delivery layer, where framing information is added, and the resulting data is transported to the network or the storage device.

The watermark $w(x, y)$ is added to the luminance plane of the VOPs. Since the DCT is a linear transform, adding the transformed watermark signal directly to the DCT coefficients of the luminance blocks is equivalent to addition in spatial domain. Hence, the elementary bitstream is parsed partially and only the DCT coefficients are modified. All other information is retained and later used to re-assemble the watermarked bitstream (see Fig. 2).

An elementary bitstream is parsed down to the block level and variable-length coded motion vector and DCT coefficients are obtained. Motion vectors are reconstructed by VLC decoding and reversing any prediction steps when applicable. Likewise, VLC decoding, inverse zig-zag scanning, inverse prediction, and de-quantization are employed to obtain DCT coefficients. After the watermark signal is embedded, VLC codes corresponding to the DCT coefficients are regenerated and the bitstream is reconstructed.

Insertion of the watermark signal into the reconstructed DCT coefficients is illustrated in Fig. 3. Before adding the watermark $w(x, y)$ to the VOPs, $w(x, y)$ is divided into 8×8 nonoverlapping blocks, transformed to the DCT domain, and the DC coefficient of each block is set to 0. The latter step maintains the integrity of the bitstream by preserving the original direction of the DC prediction. Removing the DC terms often has an insignificant effect on the overall performance of the algorithm. The transformed watermark $W(u, v)$ is added to the DCT coefficients of the luminance blocks of a VOP as follows.

For every luminance block of a given VOP:

- 1) Decode the DCT coefficients by decoding the VLC codes, converting the run-value pairs using the given zig-zag

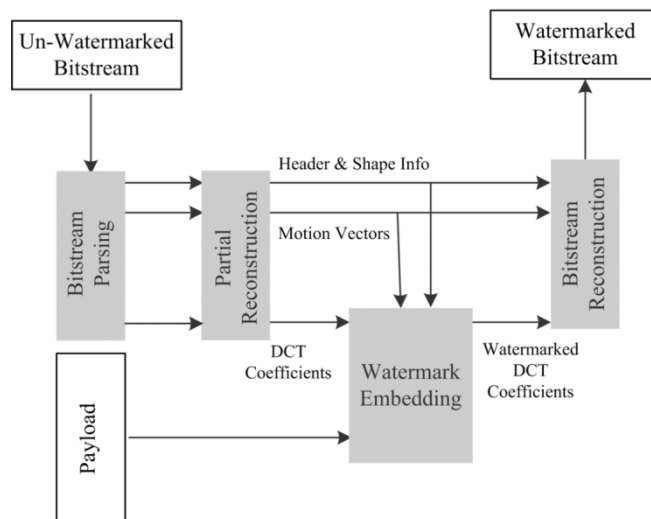


Fig. 2. Basic steps of the compressed-domain watermarking method.

scan order, reversing the AC prediction (if applicable), and inverse quantization using the given quantizer scale.

- 2) Obtain the part of the $W(u, v)$ corresponding to the location of the current block $mod N$ in the horizontal direction and $mod M$ in the vertical direction.
- 3) Scale the watermark signal by a content-adaptive local gain and a user-specified global gain.
- 4) If the block is inter-coded, compute a drift signal using the motion compensated reference error.
- 5) Add the scaled watermark and the drift signal to the original AC coefficients. Unlike [5], all coefficients in non-skipped macroblocks are considered for watermark embedding.
- 6) Re-encode the DCT coefficients into VLC codes by quantization, AC prediction (if applicable), zig-zag scanning, constructing run-value pairs, and VLC coding.
- 7) If necessary, selectively remove DCT coefficients and redo the VLC coding to match the target bit-rate.
- 8) Adjust the coded-block pattern to match the coded and uncoded blocks properly after watermarking.

Once all the blocks in a VOP are processed, the bitstream corresponding to the VOP is re-assembled. Hereunder, we detail the gain adaptation, drift compensation, and bit-rate control procedures.

a) Adaptive Gain (Local Gain Control): The objective of the adaptive gain (or local gain control) is to improve the performance of the watermark by adapting the watermark embedding to the local characteristics of the host video. For relatively “smooth” regions of the video, where even a small amount of distortion may be visible, the local gain control reduces the watermark embedding power to minimize watermark perceptibility. For relatively “busy” or textured regions of the image, the local gain control increases the embedding power for improved robustness. The gain control is constrained by computational limits to preserve the advantage of compressed-domain watermark embedding and may not be able to exploit features that require expensive analysis, such as multichannel visual modeling or temporal masking.

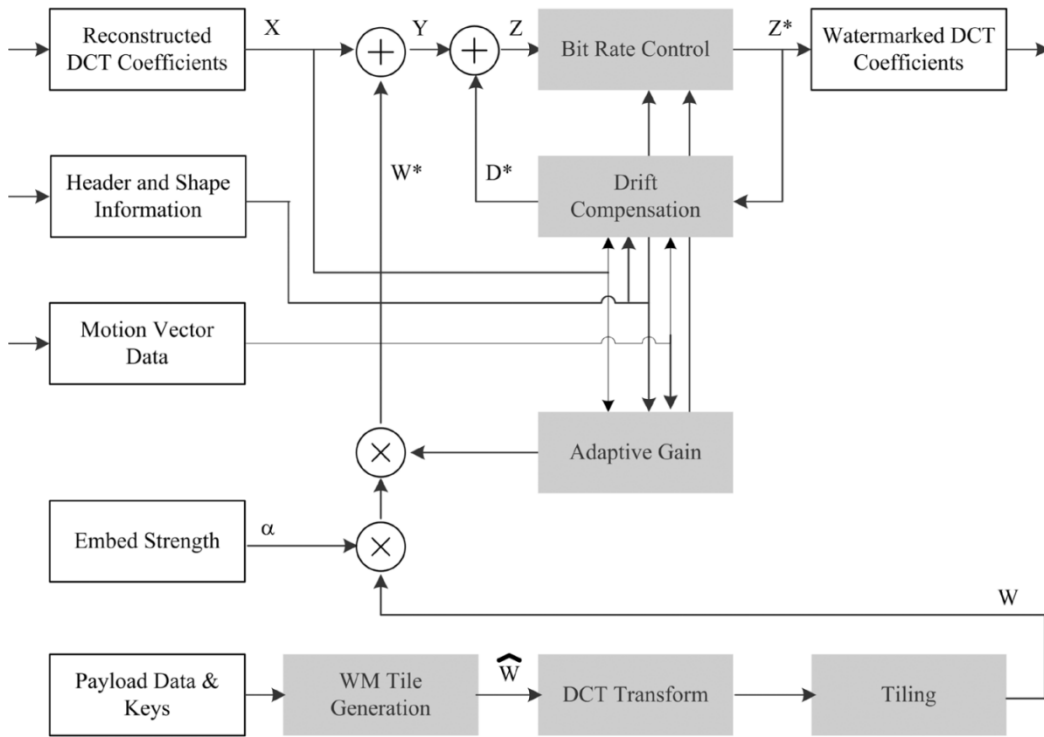


Fig. 3. Watermark generation and insertion to the reconstructed DCT coefficients.

In Hartung's adaptive gain method [5], watermark coefficients are scaled in proportion to the corresponding DCT coefficient where the watermark coefficient will be embedded (with possible thresholding). Arena [30] describes another method that weights the watermark embedding power based loosely on a visual model [31]; however, the technique was applied only to the intra-frames of MPEG-2 and predicted frames (P- and B-frames) were not watermarked.

Our local gain control method is applicable for both intra-coded VOPs as well as predicted VOPs. Our method uses a local activity measure to adjust the watermark power on a block-by-block basis, which is obtained directly from the DCT coefficients for intra-blocks and predicted using motion-vector information for predicted blocks. It does not require the video to be fully decoded and is computationally efficient.

Fig. 4 shows our local gain control model. Information about the video, such as the DCT coefficients and motion vector data, are provided to a gain model. The gain model outputs local gain weights $L(x, y)$, where (x, y) refer to spatial coordinates in the video frame. The watermark coefficients are then weighted by $L(x, y)$ to produce the watermark signal that will be embedded into the video

$$W^*(x, y) = \alpha L(x, y)W(x, y) \quad (6)$$

where W^* is the watermark that will be embedded, α is the user-selected global gain, and W is the watermark signal prior to gain adjustment. As a special case, disabling the adaptive gain is the equivalent of selecting $L(x, y) = 1.0$ for all (x, y) .

Our gain model assigns local gain weights on a block-by-block basis, with each block corresponding to a single block in MPEG-4 (8×8 pixels in size.) For each VOP, two steps are performed: 1) *activity estimation*, which estimates the amount

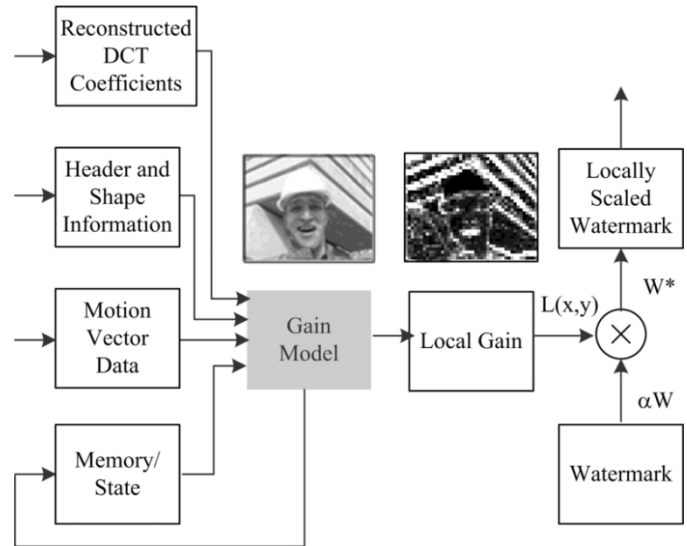


Fig. 4. Adaptive gain model.

of spatial activity (busy-ness) for each block, followed by *weight assignment*, which determines the local gain weights based on the estimated activity in the VOP. Because the encoded video data are different for intra-coded VOPs and predicted VOPs (P-VOPs and B-VOPs) in MPEG-4, two different methods are used for estimating activity. For all blocks in I-VOPs and intra-coded blocks occurring in predicted VOPs, the energy of the DCT coefficients (which is related to the variance of the spatial pixel values) is used as an estimate of activity

$$A_i = \sum_{k=\text{start}}^{\text{end}} [\text{DCT}_k]^2 \quad (7)$$

where A_i is the activity measure of block i , i is the block index, and DCT_k is the reconstructed value of the k -th DCT coefficient in zig-zag order (DCT_0 is the DC coefficient). As described, this calculation requires inverse intra-AC prediction to be performed on each block. It may be useful to select a *start* value other than 1, to prevent strong edges or text in the video from influencing the activity estimate too greatly.

For nonintra-blocks in predicted VOPs, (7) is not an appropriate activity estimator because the encoded DCT values in the bitstream represent the prediction residual from motion compensation and not the base-band image itself. High DCT coefficient values in these blocks indicate temporal prediction is performing poorly and does not necessarily indicate high spatial activity or busy-ness. One method for activity estimation would be to decode and reconstruct all predicted VOPs fully and then use (7) on each block. We adopt another method that uses motion vector information to estimate the activity of predicted blocks in a manner similar to motion compensation.

Our method for activity estimation in predicted blocks requires the local gain control to memorize the activity estimates of blocks in previously decoded VOPs, analogous to the picture buffers used by MPEG-4 decoders for motion compensation. Unlike the motion compensation, only a single value is retained for each block. The estimated activity of each predicted block is then an average of the estimated activity of blocks in the reference frame(s), weighted appropriately by motion vector information as shown in Fig. 5. Note that the activity measure A_i , given by

$$A_i = \left(\frac{N_1}{N}\right) A_A + \left(\frac{N_2}{N}\right) A_B + \left(\frac{N_3}{N}\right) A_C + \left(\frac{N_4}{N}\right) A_D \quad (8)$$

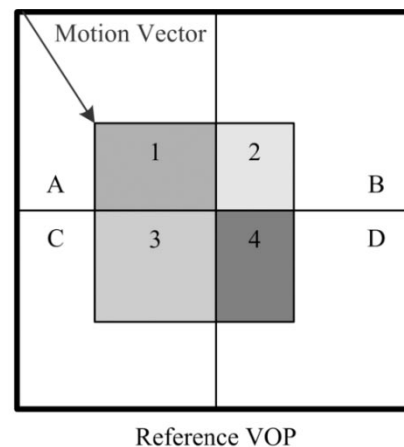
ignores the DCT values of the encoded residual for the predicted block to obtain A_i . The computation of (8) only requires the motion vector to be decoded for the predicted block and has little computational cost compared with motion compensation and VOP reconstruction.

Once the activity estimates (A_i s) for all blocks in the current VOP have been obtained, the local gain weight for each block is

$$L_i = \frac{\sqrt{A_i}}{\frac{1}{M} \sum_{k=1}^M (\sqrt{A_k})} = \frac{\sqrt{A_i}}{\text{mean}(\sqrt{A_k})} \quad (9)$$

where L_i is the local gain weight for block i , A_i is the activity estimate for block i , i and k are block indices, and M is the total number of blocks in the VOP. Equation (9) gives greater weight to blocks with higher activity estimates in the VOP, which causes the watermark to be embedded more strongly in busy regions of the VOP while at the same time attenuating the watermark in relatively smooth regions of the VOP. The local gain weights may also be thresholded to within a desired range, preventing outliers from affecting the visual quality too greatly, and preventing blocking artifacts.

b) Drift Signal Compensation: In compressed domain watermarking, when the watermark signal is inserted to a frame, it “leaks” into successive frames that use that frame as a reference in temporal prediction (motion compensation). If not properly compensated, a drift between the intended reference at the encoder and the reconstructed reference at the decoder is



$$A_i = \left(\frac{N_1}{N}\right) A_A + \left(\frac{N_2}{N}\right) A_B + \left(\frac{N_3}{N}\right) A_C + \left(\frac{N_4}{N}\right) A_D$$

A_i = Estimated Activity of block i in current VOP

N_1 = # of Pixels in Area 1

N_2 = # of Pixels in Area 2

N_3 = # of Pixels in Area 3

N_4 = # of Pixels in Area 4

$N = N_1 + N_2 + N_3 + N_4 = 64$

A_A = Estimated Activity of block A in reference VOP

A_B = Estimated Activity of block B in reference VOP

A_C = Estimated Activity of block C in reference VOP

A_D = Estimated Activity of block D in reference VOP

Fig. 5. Activity estimation for predicted blocks.

formed. Drift in watermarking applications has two different, but related effects: 1) leaking watermark signal interferes with the watermark signal that is embedded in the consecutive frames and 2) accumulation of drift error may cause visual artifacts and may become intolerable. Inter-frame interference may be constructive and improve watermark detection. This phenomenon is frequently observed when there is uniform or no motion between consecutive frames. Nevertheless, the motion field is often nonuniform and the interference is destructive. That is, motion vectors within a frame are in different directions and scramble the previous watermark signal, preventing constructive interference. Bearing similar characteristics to the real watermark, the scrambled signal often hinders detection and deteriorates the performance.

Here, a spatial-domain drift compensator similar to that of [5] is employed to cope with unwanted interference and visual degradations. In particular, a drift compensator keeps track of the difference between the unwatermarked reference VOP at the encoder and watermarked reference VOP that will be reconstructed at the decoder. Before watermarking an inter-coded VOP, the error signal from the previous VOP is propagated via motion compensation and subtracted from the current VOP. At each VOP, the error signal is updated to include the latest modifications made within the current VOP. Feeding back the changes, the system appropriately tracks the difference between the states

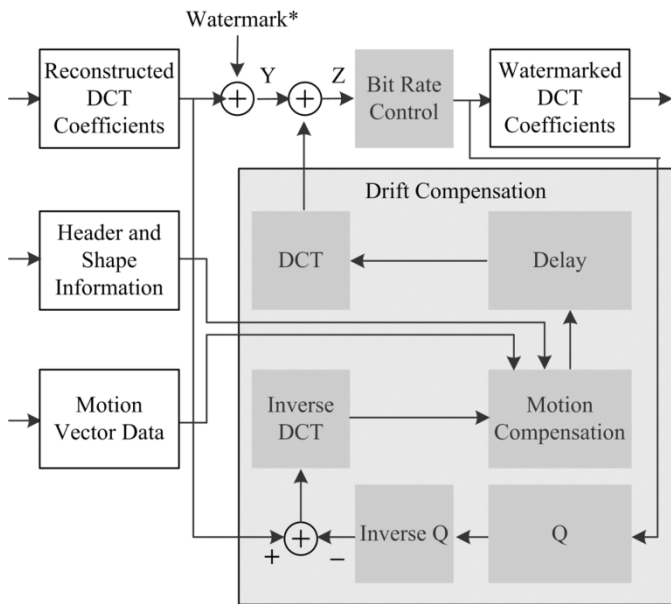


Fig. 6. Drift between the encoder and decoder due to watermarking is compensated by a feedback loop.

of the encoder and the decoder, even in the presence of quantization noise. A block diagram for the drift compensator is shown in Fig. 6. Note that the motion compensation may be performed directly in DCT domain as described in [5], eliminating the transform overhead and reducing computational requirements.

c) Bit-Rate Control: Spread-spectrum watermarks often contain substantial mid- to high-frequency contents to limit the interference from the host signal whose energy is concentrated in low- to mid-frequency bands. Nevertheless, this mismatch between signal characteristics creates a challenge for the compressed domain representation of the watermarked signal. Often, watermarked video consumes substantially more bits than unwatermarked video. Although, a small increase in bit rate may be tolerable in some applications, in general, a bit-rate control algorithm is needed to keep the bit rate within the pre-specified limits.

In [5], Hartung controls the data rate of the watermarked bit-stream by skipping modification (watermarking) of a DCT coefficient whenever such a modification increases the bit rate over a preset limit. This approach scales back, or sacrifices, the watermark to meet the bit-rate constraints. In low bit-rate applications, where only a fraction of the few nonzero coefficients can be modified, this technique severely limits the robustness of the watermark.

Herein, we take an alternative approach and allow for the parts of original signal to be removed in favor of a more robust watermark. In our approach, first the watermark signal is added to all (not only nonzero) AC coefficients of the block, and then the quantized DCT coefficients of the modified block are selectively eliminated (set to 0) to meet the target bit rate. In each turn, the quantized DCT coefficient with the minimum absolute value is eliminated until the remaining coefficients can be represented within the target bit rate. This process decreases the number of nonzero DCT coefficients, and eventually reduces the number of bits required. Note that the elimination process

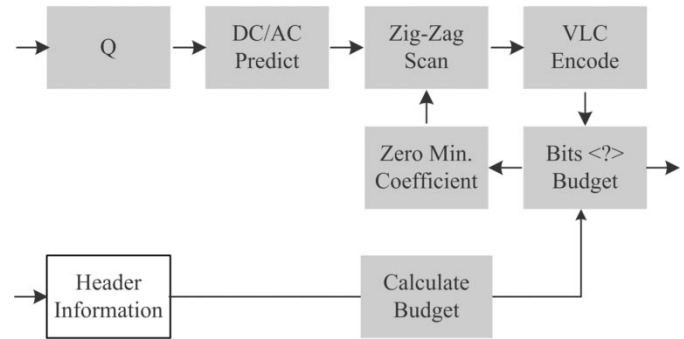


Fig. 7. Bit-rate control.

does not differentiate between the host signal and the watermark signal. As a result, in some instances, it sacrifices the host signal quality instead of limiting the amount of embedded watermark. This property is especially useful for embedding a robust watermark in lower bit-rate applications, where only a few coefficients can be marked using the technique proposed in [5].

Bit allocation is a challenging problem in compression and various optimization methods have been developed [32]. Herein, the problem is revisited in the watermarking context, where the fidelity of the embedded watermark signal has to be traded with that of the host signal through bit allocation. That is, we seek to determine the best allocation of available bits between different watermarked image blocks. We now introduce two heuristic approaches and defer the theoretical optimization problem for future research. We denote the permitted overall rate increase by R , the number of bits used by a DCT block before watermarking by B_{orig} , and the bits allocated by the algorithm, i.e., bit-budget or target rate, by B_{new} .

The first method is a simple strategy that piggybacks onto the encoder's bit-rate control algorithm, and it allocates bits in proportion to original number of bits. In particular, for each block

$$B_{\text{new}} = B_{\text{orig}}(1 + R) + \Delta \quad (10)$$

where Δ is the number of bits that have been assigned to previous blocks but have not been used. In some instances, the watermarked block requires fewer bits than allocated through this algorithm and the Δ term in (10) provides the necessary feedback for better utilization. Typically, the bit-rate controller of the encoder, e.g., TM5, allocates more bits to textured areas of the VOP [32]. As a result, greater numbers of additional bits are allocated for these textured areas, which in turn allows for a more accurate (robust) representation of the watermark. This behavior is in agreement with the local gain adaptation algorithm, which calls for stronger watermarks by increasing the gain in textured areas (see Section III-B).

A more elaborate and flexible scheme is obtained when the bit-rate controller explicitly observes the desired watermark strength and the default rate increase due to watermark embedding for each block. In this method, the target rate for the current block is determined by

$$B_{\text{new}} = B_{\text{orig}} + \left(\frac{\text{Increase} + \text{Allocation}}{2} \right). \quad (11)$$

Equation (11) seeks to strike a balance between the default increase due to watermark addition and an allocation of remaining bits based on the local gain factor. In particular, the default increase in a block's rate after watermark addition is given by

$$\text{Increase} = B_{\text{watermarked}} - B_{\text{orig}} \quad (12)$$

where $B_{\text{watermarked}}$ denotes the number of bits required by the watermarked block before coefficient elimination. On the other hand, Allocation is the part of the remaining additional bits (T_{remain}) that is proportional to the local watermark gain factor (G_{local})

$$\text{Allocation} = \frac{G_{\text{local}}}{\sum G_{\text{local}}} T_{\text{remain}} \quad (13)$$

where summation is over all remaining blocks in the VOP.

T_{remain} is initialized by the total increase for the current VOP, i.e., $\sum B_{\text{orig}} R$, and it is updated once a block is written to the output stream. Although it is possible to assign a new target using the Allocation term only, the Increase term in (11) provides additional flexibility and lets a particular block consume more bits than otherwise available with the Allocation. Note that such an occasional over-consumption consequently reduces the number of remaining bits (T_{remain}), hence it does not affect the overall bit rate. Likewise, if the actual rate of the block is less than the target rate, unused bits remain in T_{remain} and are utilized subsequently. Note that this method spreads these unused bits over all remaining blocks of the VOP, where the earlier method made them available immediately for the next block regardless of the local gain. The latter method's flexibility and its direct dependence on local gain values results in a better tradeoff between host signal quality and embedded watermark strength. In our experiments, we have observed consistently better overall visual quality and/or better watermark robustness with the later method.

2) *Watermark Detection*: Since a spatial watermark was used, watermark detection is performed after decompressing the bitstream. Adding a DCT transformed version of the watermark to the DCT coefficients of the image in the compressed domain is similar to adding the nontransformed watermark to pixels of the image in the spatial domain (the only difference is the effect of quantization). Detection is performed on the luminance component in two steps for each VOP: First, the detector is synchronized by resolving the scale and orientation. Next, the watermark message is read and decoded.

The scale and orientation of the VOP is resolved using $g(x, y)$ and the log-polar re-mapping described in Section III-A, as follows.¹ First, the VOP is divided into blocks of size NxM , and then all the blocks with fair amount of details are selected for further processing. All areas outside the boundary of a VOP are set to 0. This selective processing of the blocks enhances SNR and reduces the processing time. The SNR can be further enhanced by predicting the host image data and subtracting the prediction from the VOP. Next, average magnitude of the FFT of all these blocks is computed and used to calculate the re-map-

ping described in (4). Finally, the linear shifts in (4) are detected using a POM filter using the log-polar transform of $g(x, y)$. The calculated scale and orientation are used to invert the geometrical transformation of each NxM block. The origin of the watermark in each NxM block is calculated by matching the FFT of the block to the FFT of the sync signal using a POM filter. Once the geometric transformation and the origin of the watermark are resolved, a linear correlator can be used to read the watermark. Then, the message is obtained by error correction decoding.

IV. IMPLEMENTATION AND RESULTS

A. Test Setup

Our algorithm was tested with the first 5 s of the standard sequences: *Foreman*, *Flower Garden*, *Football*, and *Salesman*. All sequences were encoded with MPEG-4 at 128 kb/s (QCIF 176×144), 384 kb/s, and 768 kb/s (CIF 352×288) at 15 frames/s. Resulting bitstreams are supported under ASP and selected bit rates are in accordance with ASP levels L0–L3. The sequences were encoded as a single rectangular video object. The GOV structure was comprised of an I-VOP followed by 14 P-VOPs, which corresponds to one I-VOP per second.

The distortion (PSNR) between the luminance channels of the original (uncompressed) sequence and the compressed-but-not-watermarked and compressed-and-watermarked sequences was computed. Δ PSNR value, which signifies the ratio of distortions due to compression and watermarking, [5] was derived. Detection results are represented by two metrics. The first metric is the per-frame detection rate and indicates the ratio of frames where the watermark is detected and all bits are correctly decoded. (The detection was performed independently on each VOP.) Per-second detection rate is the second metric, and it is derived from per frame detection decisions by looking for detections in a sliding window of 15 frames (1-s period). Per-second detection rate is meaningful for applications that require at least one detection within a given interval. It also differentiates between bursts of detections versus consistent detections.

Robustness of the algorithm was tested in five categories: de-compression only (no attack), filtering, scaling, rotation, and transcoding. Filtering operations included 3×3 Gaussian and unsharp masking (Matlab default parameters), and Gamma correction ($\gamma = 0.8$). Scaling operations included scaling in spatial dimensions with factors of 75%, 90%, 110%, and 125%, and rotation was performed for 1° , 3° , and 5° (with bilinear sampling). In transcoding,² bitstreams were decompressed and re-compressed at the same bit rate using a different GOV structure (I-VOP followed by 19 P-VOPs).

B. Experimental Results

All test sequences were watermarked using the proposed method and two different global embedding strengths, which were determined empirically. Local gain control, drift-compensation, and bit-rate control algorithms were turned on and

¹In the current implementation, only the template imposed by the embedding of the synchronization signal $g(x, y)$ is used for synchronization. The treatment of the other template is similar, such that the autocorrelation of the whole VOP—computed using FFT—is replaced with the FFT magnitude of the blocks.

²Limited capabilities of the available MPEG-4 encoder prevented us from transcoding the 768 kb/s sequences.

TABLE I
SEQUENCES BEFORE AND AFTER WATERMARKING

Sequence	Unmarked		Marked			Detection Percentage (per frame and per second)									
	Bit Rate (kb/s)	PSNR (dB)	ARate (%)	PSNR (dB)	AFSNR (dB)	No Attack		Filtering		Scaling		Rotation		Transcoding	
						/Frame	/Second	/Frame	/Second	/Frame	/Second	/Frame	/Second	/Frame	/Second
Flower Garden	128	27.15	4	25.63	-1.52	4	53	5	69	4	52	4	53	5	77
			9	25.25	-1.9	11	100	9	100	8	88	8	94	9	100
	384	26.57	0	25.61	-0.96	4	52	4	44	4	47	4	43	3	28
			2	25.33	-1.24	7	77	6	69	7	83	6	77	5	77
	768	29.93	10	28.6	-1.33	13	100	11	91	8	90	8	96	N/A	N/A
			10	27.88	-2.05	29	100	26	100	21	100	22	100	N/A	N/A
Avg.	27.88	5.8	26.38	-1.50	11.3	80.3	10.2	78.8	8.7	76.7	8.7	77.2	5.5	70.5	
Football	128	27.89	-2	26.9	-0.98	4	27	4	37	3	27	3	27	3	27
			0	26.55	-1.33	12	100	9	100	8	79	6	69	8	80
	384	28.69	-2	28.15	-0.54	7	27	8	27	6	27	7	27	3	8
			0	27.81	-0.88	20	100	18	100	12	95	12	84	11	62
	768	31.97	2	31.15	-0.82	7	100	5	76	6	83	5	68	N/A	N/A
			6	30.63	-1.35	47	100	41	100	37	100	36	100	N/A	N/A
Avg.	29.52	0.7	28.53	-0.98	16.2	75.7	14.2	73.3	12.0	68.5	11.5	62.5	6.3	44.3	
Foreman	128	33.84	0	32.57	-1.27	5	77	4	51	3	45	2	34	4	52
			8	31.75	-2.08	12	100	10	100	8	76	8	92	9	77
	384	35.16	0	34.3	-0.86	8	100	6	89	5	70	6	92	4	27
			9	33.53	-1.63	25	100	22	100	16	100	14	100	17	100
	768	38.3	10	36.61	-1.7	41	100	37	100	27	100	26	100	N/A	N/A
			10	35.5	-2.81	67	100	64	100	52	100	45	100	N/A	N/A
Avg.	35.77	6.2	34.04	-1.73	26.3	96.2	23.8	90.0	18.5	81.8	16.8	86.3	8.5	64.0	
Salesman	128	36.86	0	34.87	-1.99	25	92	24	84	14	59	16	79	27	90
			2	33.95	-2.91	67	100	63	100	63	100	68	100	75	100
	384	37.28	0	35.98	-1.3	73	100	54	100	38	89	46	100	76	100
			2	35.3	-1.98	100	100	96	100	86	100	92	100	99	100
	768	40.24	2	38.55	-1.69	47	100	30	77	18	75	28	97	N/A	N/A
			3	37.39	-2.85	100	100	100	100	98	100	100	100	N/A	N/A
Avg.	38.13	1.5	36.01	-2.12	68.7	98.7	61.2	93.5	52.8	87.2	58.3	96.0	69.3	97.5	
Overall	32.82	3.5	31.24	-1.58	30.63	87.7	27.3	83.9	23.0	78.5	23.8	80.5	22.4	69.1	

a 10% increase in the bit rate was allowed. The start and end values used in the adaptive gain activity estimation [(7)] were 10 and 63, respectively, with the start value chosen empirically to prevent strong edges from influencing the activity estimation too greatly.

Table I shows the performance of the technique for all “attacks”. On average, the watermarking process increased the size of the compressed bitstream by 3.5%, whereas the PSNR of the compressed sequence was decreased by 1.6 dB. It was observed that this amount of degradation is more tolerable visually at 768 and 384 kb/s than at 128 kb/s. In the case of the *Flower Garden* and *Football*, the quality of the watermarked video was evaluated as acceptable at 768 kb/s but was objectionable at 128 kb/s. (These observations are further validated by the subjective tests, see Section IV-B-1.) This degradation can be attributed to the fact that at lower data rates, the compressed bitstream carries only visually significant features of the video. Modifying these features during the watermark embedding process creates significant distortion. However, at higher data rates, the watermark can be embedded into visually less significant features. Thus, maintaining video quality after watermarking at lower data rates is more challenging.

For all test sequences, the watermark was decoded correctly on average from more than 30% of the frames with no attack and from more than 20% of the frames under various manipula-

tions. In a given 1-s interval, these detection rates translated to a success rate of approximately 90% and 80%, respectively.

In general, higher detection rates were obtained at 768 and 384 kb/s rather than at 128 kb/s. Moreover, CIF video detected better than QCIF video, because CIF images provided more data to the averaging processes used to calculate the sync signal (see Section III-A). It was also observed that the watermark detection rates were higher for the *Football* and *Salesman* sequences. These sequences have little or no global motion and the moving objects are limited to relatively small regions of the frame. In these sequences, the watermark leaks from the I-VOP to the consecutive P-VOPs due to the temporal prediction in compression. The phase of the global synchronization signal is not disturbed by the local motion and insufficient drift compensation, resulting in a higher detection rate. Note that, as the watermark and drift signals cancel each other, no modification is necessary for the P-VOPs. Hence, the data rate increase for these sequences is relatively small.

1) *Subjective Quality Test Results:* To assess the visual effects of the watermarking method subjectively, we ran an informal test with a small number of subjects. In this nonblind test, nine subjects were shown the original and three watermarked (with different embedding strengths) versions of each sequence and asked to rate the distortion they perceived. The responses were gathered according to the scale shown on Table II. Mean

TABLE II
SUBJECTIVE TEST RESPONSES

Not Noticeable	5
Almost Noticeable	4
Acceptable	3
Somewhat Objectionable	2
Objectionable	1

TABLE III
SUBJECTIVE TEST SCORES

Rate (kb/s)	Sequence				
	<i>Flower Garden</i>	<i>Football</i>	<i>Foreman</i>	<i>Salesman</i>	Avg.
128	2.7	2.1	3.5	3.4	2.9
	1.9	1.5	2.8	2.3	2.1
384	2.8	2.8	4.0	4.0	3.4
	2.4	2.2	3.4	3.5	2.9
768	3.1	3.9	4.4	4.6	4.0
	2.9	3.0	4.1	3.9	3.5
Avg.	2.6	2.6	3.7	3.6	

Average subjective test scores of each watermarked bitstream and bit rate and sequence averages.

response from all subjects for the two embedding strengths, for which the detection results were reported, is seen in Table III.

Subjective quality results first validate the fundamental tradeoff between the increase in quality distortion and increased watermark strength, and thus improved detection performance. Upon inspection of different bit rates, it was seen that the subjects find the watermark more objectionable whenever the quality of the underlying compressed bitstream is lowered. That is, it is more challenging to insert imperceptible/unobjectionable watermarks at lower bit rates. This observation further reinforces the difficulty encountered from the watermark detection perspective. Sequences that contain relatively less motion, i.e., *Foreman* and *Salesman*, were regarded as more acceptable. This result may be attributed to the higher quality of unwatermarked compressed sequences in accordance with the earlier observations. Note that, increased temporal redundancy in these sequences yields to better quality at a given bit rate.

Interviews with the subjects revealed that the watermark signal was more visible over fast moving regions of the frame. The local gain adaptation algorithm presented in this paper does not account for temporal masking attributes of the human visual system. Moreover, accuracy of the energy estimation algorithm within the gain calculation degrades in the existence of fast motion. Thus, the local gain values deviate from their ideal values. Errors in the gain adjustment are further emphasized by the motion blur in these areas. Motion blur filters the high spatial frequencies that normally mask the watermark signal.

2) *Performance Improvement With Local Gain Control*: Evaluating the performance of the local gain control is challenging because of the difficulty in finding an objective visual distortion measure for examining the visual distortion for low bit rate, watermarked video. It is well known that the mean-square error and PSNR may not account for the human perceptual sensitivity to the distortion between two images or



(a)



(b)

Fig. 8. VOP of watermarked *Flower Garden* sequence (384 kb/s CIF) with adaptive gain disabled and enabled. (a) Adaptive gain disabled. (b) Adaptive gain enabled.

videos [33]. Fig. 8 shows watermarked VOPs from the *Flower Garden* sequence with adaptive gain turned on and off. When enabled, the adaptive gain reduces the power of the watermark in the smooth areas (the sky) and increases the power in busy areas (the flowers.) Subjectively, the watermark was much less visible when the adaptive gain is enabled at the same PSNR; however, even after examining [31], [33]–[38], it was difficult finding an objective measure that consistently showed the same conclusion as subjective quality observations. Finding a good objective quality measure for compressed, watermarked video is an open problem.

In our experiment, we used the Universal image quality metric described in [37] because it showed reasonable correlation with subjective quality during empirical testing using the *Flower Garden*, *Foreman*, *Football*, and *Salesman* videos. The Universal metric takes on values between 0.0 and 1.0, with higher values indicating that the images being compared are

TABLE IV
VISUAL QUALITY AND DETECTION RATES

Sequence		Bit-Rate Limit +0%			Bit-Rate Limit +10%		
		PSNR (dB)	%Detect /Frame	%Detect /Second	PSNR (dB)	%Detect /Frame	%Detect /Second
<i>Flower Garden</i>	128	25.49	4	52	25.63	4	53
	384	25.59	5	45	25.61	4	52
	768	27.71	8	92	28.60	13	100
	Average	26.26	5.7	63.0	26.60	7.0	68.3
<i>Football</i>	128	26.87	4	27	26.90	4	27
	384	28.15	7	27	28.15	7	27
	768	30.93	4	77	31.15	7	100
	Average	28.65	5.0	43.7	28.70	6.0	51.3
<i>Foreman</i>	128	32.56	4	52	32.57	5	77
	384	34.29	8	100	34.30	8	100
	768	36.20	9	100	36.61	41	100
	Average	34.35	7.0	84.0	34.50	18.0	92.3
<i>Salesman</i>	128	34.87	24	95	34.87	25	92
	384	35.95	68	100	35.98	73	100
	768	38.42	32	100	38.55	47	100
	Average	36.41	41.3	98.3	36.50	48.3	97.3
Overall		31.42	14.8	72.3	31.60	19.8	77.3

Visual quality and detection rates (no attack case) when bit rate increase is limited by 0% (no increase) and 10%. Embedding strengths, thus 10% results, are identical to odd lines in Table I.

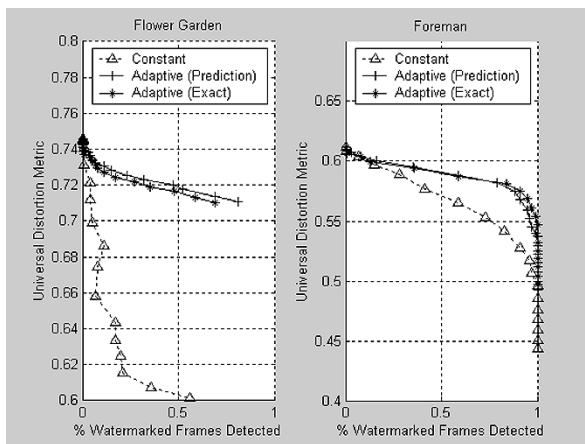


Fig. 9. Adaptive gain performance for *Flower Garden* and *Foreman* sequences.

more perceptually similar. The Universal metric indicated decreasing perceptual quality as the global watermark embedding strength was increased as well as decreased quality for lower bit-rate videos. However, it is realized that the Universal metric is a still-image metric that does not account for temporal effects of visual perception.

The global gain parameter was varied and the corresponding visual quality (mean Universal metric value across all frames) and the per-frame detection rate were measured. The performance of the adaptive gain for the *Flower Garden* and *Foreman* sequences is shown in Fig. 9. Three sets of results are shown: Adaptive gain disabled (constant gain), adaptive gain using full-frame reconstruction and (7) for all blocks in all VOPs (adaptive gain-exact), and adaptive gain using temporal prediction for activity estimation in predicted blocks as described in Section III-B-1 (adaptive gain-prediction).

Fig. 9 shows that for any fixed detection rate, the use of the adaptive gain allows improved visual quality than that of constant gain. The subjective quality agrees with the Universal metric for these two video sequences and noticeable improvement can be observed when the adaptive gain is enabled, particularly for the *Flower Garden* sequence. The graph also shows very little difference between the performance of the adaptive gain when motion vector and temporal prediction is used for estimating activity in predicted blocks, as compared to using full-frame reconstruction. However, performance of the temporal prediction, thus of the adaptive gain, may degrade significantly when a shot boundary occurs at a P-VOP. In addition, some subjects noticed a slight flicker between the last P-VOP of a GOV and the I-VOP of the next GOV when adaptive gain with temporal prediction was used. In the existence of motion, the activity estimate from temporal prediction degrades gradually. At an I-VOP, the activity estimate suddenly is corrected using the calculation in (7). The sudden change is often observed as flicker, particularly at high embedding strengths. With the exception of these cases, the visual quality of the watermarked video using the adaptive gain shows dramatic improvement.

The weakness of the adaptive gain control is that it is based solely on spatial activity of the video. The temporal characteristics of human perception are not accounted in our gain model, which can give rise to artifacts such as very slight “mosquito” effects and flickering in the watermarked video. These effects are somewhat masked by the quantization and compression noise present in the unwatermarked video; however, subjective test results confirm that the watermark is most visible in areas of motion, which is clearly a temporal phenomenon.

3) *Effects of Limited Bit Rate and Bit-Rate Control*: As stated earlier, size of the watermarked bitstream poses another limitation for compressed domain watermarking. As the bit-rate

TABLE V
AVERAGE DETECTION RATES

Sequence	Detection Percentage (%)							
	$N=1$		$N=3$		$N=5$		$N=15$	
	/Detect	/Second	/Detect	/Second	/Detect	/Second	/Detect	/Second
<i>Flower Garden</i>	11.3	80.3	33.7	81.8	50.0	85.2	92.5	90.8
<i>Football</i>	16.2	75.7	29.5	85.8	36.2	79.0	68.0	63.3
<i>Foreman</i>	26.3	96.2	46.2	89.8	49.5	77.8	65.2	66.2
<i>Salesman</i>	68.7	98.7	71.2	98.0	71.2	95.8	82.0	86.0
Overall	30.6	87.7	45.1	88.9	51.7	84.5	76.9	76.6

Average detection rates when N frames are averaged before detection.

control mechanism eliminates DCT coefficients, which represent the host and/or the watermark signal, often both the visual quality of the video and the watermark detection performance are degraded (see Table IV). In contrast to earlier systems, the system trades off the quality to achieve better detection, under the data rate constraints. A system that sacrifices only the watermark to control the data rate would provide more limited detection performance.

4) *Frame Accumulation for Robust Detection*: In the experiments presented so far, watermark detection was performed on each individual frame (VOP). Since the same watermark was inserted in each frame in a GOV, watermark signal contained a significant temporal redundancy, which may be exploited for improved detection performance. Here, frames within a sliding window of size N were averaged and the average frame was used for detection.

In our experiments, N was set to 1, 3, 5, and 15. In Table V, we observed that the success rate for each detection increased significantly through this method (from 30.6% to 76.9%). Nevertheless, the percentage of 1-s intervals where a watermark was successfully detected (per second detection rate) did not necessarily improve. Upon close inspection of results, it was observed that often—especially for small N —a single frame within the sliding window forced a detection. As all window positions that include said frame were detected successfully, the success rate increased without improving per-second detection results. Despite the lack of improvement in per-second detection, detection after averaging is a useful tool that can decrease the computational requirements of a system. Averaging is a rather computationally inexpensive operation when compared with the watermark detection process. Detecting on averaged frames decreases the number of detections performed to find the first detection. Since it is often sufficient to obtain a single detection, this approach significantly reduces the computational requirements of the watermark detector. In exchange, a buffer of size N frames is required.

V. CONCLUSIONS

A technique for watermarking MPEG-4 low-bit-rate compressed bitstreams was developed and implemented. The technique requires bitstream parsing and partial decoding, but it avoids full decoding and re-encoding of the bitstream, which may be impractical for many applications. The technique features a new computationally inexpensive method for adjusting the gain of the watermark according to video characteristics, which improves the visual quality of the watermarked video. In

addition, a novel method for controlling the data rate of the watermarked video was developed that is suitable for low bit-rate video. Drift compensation prevents the accumulation of prediction error introduced by watermarking and supports the prediction modes available in MPEG-4, including intra-DC/AC prediction.

In general, watermarking of video compressed at less than 1 Mb/s is more challenging than watermarking at higher video bit rates. Test results indicated that watermarking video compressed at bit rates below 1 Mb/s may cause a small increase in video bit rate, and attempting to watermark video compressed at 128 kb/s may produce objectionable video quality degradation. Nonetheless, test results indicated that our watermark could be detected after decompression, filtering, scaling, rotation, and trans-coding. They also indicated that our method has at least 80% average detection rate based on frame moving average with less than 5% average increase in the bit rate and at least one frame per second frame-by-frame detection rate. The extension of our watermarking technique to other MPEG-4 profiles can be achieved often with only minor modifications.

ACKNOWLEDGMENT

This work was completed at Digimarc Corporation. The authors would like to thank Mrs. K. Smith of Digimarc Corporation for her assistance in editing and preparing the final manuscript.

REFERENCES

- [1] G. Langelaar, I. Setyawan, and R. Lagendijk, "Watermarking digital image and video data: a state-of-the-art overview," *IEEE Signal Processing Mag.*, vol. 17, pp. 20–46, Sept. 2000.
- [2] M. Swanson, M. Kobayashi, and A. Tewfik, "Multimedia data-embedding and watermarking technologies," *Proc. IEEE*, vol. 86, no. 6, pp. 1064–1087, June 1998.
- [3] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proc. IEEE*, vol. 87, pp. 1079–1107, July 1999.
- [4] I. Cox, M. Miller, and J. Bloom, *Digital Watermarking*. San Francisco, CA: Morgan Kaufman, 2002.
- [5] F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video," *Signal Processing*, vol. 66, no. 3, pp. 283–301, May 1998.
- [6] F. Hartung, "Digital Watermarking and Fingerprinting of Uncompressed and Compressed Video," Ph.D. dissertation, University of Erlangen, 2000.
- [7] *Information Technology—Generic Coding Of Moving Pictures and Associated Audio Information*, International Organization for Standardization, ISO/IEC 13 818-2, 1994.
- [8] *Information Technology—Coding of Audio-Visual Objects: Video*, International Organization for Standardization, ISO/IEC 14 496-2, Oct. 1998.

- [9] *Information Technology—Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mb/s, Part 1: System; Part 2: Video; Part 3: Audio*, International Organization for Standardization, ISO/IEC 11172, 1993.
- [10] E. Lin, C. Podilchuk, T. Kalker, and E. Delp, "Streaming video and rate scalable compression: what are the challenges for watermarking?," in *Proc. SPIE Security and Watermarking of Multimedia Contents III*, vol. 4314, San Jose, CA, Jan. 22–25, 2001, pp. 116–127.
- [11] E. Lin and E. Delp, "Temporal synchronization in video watermarking," in *Proc. SPIE Security and Watermarking of Multimedia Contents IV*, vol. 4675, San Jose, CA, Jan. 21–24, 2002, pp. 478–490.
- [12] C. Lin, M. Wu, J. Bloom, I. Cox, M. Miller, and Y. Lui, "Rotation, scale, and translation resilient watermarking for images," *IEEE Trans. Image Processing*, vol. 10, pp. 767–782, May 2001.
- [13] T. Kalker, G. Depovere, J. Haitsma, and M. Maes, "A video watermarking system for broadcast monitoring," in *Proc. SPIE Security and Watermarking of Multimedia Contents*, vol. 3657, San Jose, CA, Jan. 1999, pp. 103–112.
- [14] I. Mora-Jimenez and A. Navia-Vazquez, "A new spread spectrum watermarking method with self-synchronization capabilities," in *Proc. IEEE Int. Conf. Image Processing '00*, Vancouver, Canada, Sept. 10–13, 2000.
- [15] G. Langelaar and R. Lagendijk, "Optimal differential energy watermarking of DCT encoded images and video," *IEEE Trans. Image Processing*, vol. 10, pp. 148–158, Jan. 2001.
- [16] I. Setyawan and R. Lagendijk, "Low bit-rate video watermarking using temporally extended Differential Energy Watermarking (DEW) algorithm," in *Proc. SPIE Security and Watermarking of Multimedia Content III*, vol. 4314, 2001, pp. 73–84.
- [17] A. Piva, R. Caldelli, and A. De Rosa, "A DWT-based object watermarking system for MPEG-4 video streams," in *Proc. IEEE Int. Conf. Image Processing '00*, vol. 3, Vancouver, Canada, 2000, pp. 5–8.
- [18] M. Barni, F. Bartolini, V. Cappellini, and N. Checca-cci, "Object watermarking for MPEG-4 video streams copyright protection," in *Proc. SPIE Security and Watermarking of Multimedia Contents II*, vol. 3671, San Jose, CA, Jan. 2000, pp. 465–476.
- [19] P. Bas, J.-M. Chassery, and B. Macq, "Geometrically invariant watermarking using feature points," *IEEE Trans. Image Processing*, vol. 11, pp. 1014–1028, Sept. 2002.
- [20] D. Nicholson, P. Kudumakis, and J. F. Delaigle, "Watermarking in the MPEG-4 context," in *European Conf. Multimedia Applications Services and Techniques*, Madrid, Spain, May 1999, pp. 472–492.
- [21] P. Eisert and B. Girod, "Analyzing facial expressions for virtual conferencing," *IEEE Comput. Graph. Applic.—Special Issue: Computer Animation for Virtual Humans*, vol. 18, no. 5, pp. 70–78, Sept. 1998.
- [22] S. Haykin, *Communication Systems*, 3rd ed. New York: Wiley.
- [23] I. Cox, J. Killian, T. Leighton, and T. Shamoan, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Processing*, vol. 6, pp. 1673–1687, Dec. 1997.
- [24] J. O'Ruanaidh and G. Csurka, "A Bayesian approach to spread spectrum watermark detection and secure copyright protection for digital image libraries," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Fort Collins, CO, June 1999.
- [25] A. Herrigel, S. Voloshynovskiy, and Y. Rytsar, "The watermark template attack," in *Proc. SPIE Security and Watermarking of Multimedia Contents III*, vol. 4314, San Jose, CA, Jan. 22–25, 2001, pp. 394–405.
- [26] S. Pereira, J. Ruanaidh, F. Deguillaume, G. Csurka, and T. Pun, "Template based recovery of Fourier-based watermarks using log-polar and log-log maps," in *Proc. IEEE Int. Conf. Multimedia Computing and Systems*, vol. 1, 1999, pp. 870–874.
- [27] T. Kalker, G. Depovere, J. Haitsma, and M. Maes, "A video watermarking system for broadcast monitoring," in *Proc. SPIE Security and Watermarking of Multimedia Contents*, vol. 3657, San Jose, CA, pp. 103–112.
- [28] D. Delannay and B. Macq, "Generalized 2-D cyclic patterns for secret watermark generation," in *Proc. IEEE Int. Conf. Image Processing '00*, vol. 3, Vancouver, Canada, 2000, pp. 77–80.
- [29] J. O'Ruanaidh and T. Pun, "Rotation, translation and scale invariant digital image watermarking," in *Proc. IEEE Int. Conf. Image Processing '97*, vol. 1, Washington, DC, 2000, pp. 536–539.
- [30] S. Arena, M. Caramma, and R. Lancini, "Digital watermarking applied to MPEG-2 coded video sequences exploiting space and frequency masking," in *Proc. IEEE Int. Conf. Image Processing '00*, Vancouver, Canada, 2000.
- [31] C. J. van den Branden Lambrecht and O. Verscheure, "Perceptual quality measure using a spatio-temporal model of the human visual system," in *Proc. SPIE Digital Video Compression: Algorithms and Technologies 1996*, vol. 2668, San Jose, CA, Jan./Feb. 1996, pp. 450–461.
- [32] G. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Mag.*, vol. 15, pp. 74–90, Nov. 1998.
- [33] A. Basso, I. Dalgic, F. Tobagi, and C. Lambrecht, "Study of MPEG-2 coding performance based on a perceptual quality metric," in *Proc. 1996 Picture Coding Symp.*, Melbourne, Australia, Mar. 1996, pp. 263–268.
- [34] A. Webster, C. Jones, M. Pinson, S. Voran, and S. Wolf, "An objective video quality assessment system based on human perception," in *Proc. Human Vision, Visual Processing, and Digital Displays IV*, San Jose, CA, Feb. 1993, pp. 15–26.
- [35] S. Westen, R. Lagendijk, and J. Biemond, "Spatio-temporal model of human vision for digital video compression," in *Proc. SPIE Human Vision and Electronic Imaging II*, vol. 3016, San Jose, CA, 1997, pp. 260–268.
- [36] A. Watson, J. Hu, and J. McGowan III, "Digital video quality metric based on human vision," *J. Electron. Imaging*, vol. 10, no. 1, pp. 20–29, 2001.
- [37] Z. Wang and A. Bovik, "A universal image quality index," *IEEE Signal Processing Lett.*, vol. 9, pp. 81–84, Mar. 2002.
- [38] S. Voloshynovskiy, S. Pereira, V. Iquise, and T. Pun, "Attack modeling: toward a second generation benchmark," *Signal Processing—Special Issue: Information Theoretic Issues in Digital Watermarking*, pp. 1177–1214, June 2001.

Adnan M. Alattar (M'86) was born in Khanyounis, Palestine, in 1961. He received the B.S. degree from the University of Arkansas, Fayetteville, in 1984, and the M.S. and Ph.D. degrees from North Carolina State University, Raleigh, in 1985 and 1989, respectively, all in electrical engineering.

He was a Senior Algorithm Engineer at Intel Corporation from 1989 to 1995 and an Assistant Professor at King Fahd University for Petroleum and Minerals from 1995 to 1998. Since 1998, he has been a Senior Research and Development Engineer at Digimarc Corporation, Tualatin, OR. He holds 11 U.S. and two European patents in the area of video compression and digital watermarking and is the author of several technical papers. His areas of research interest include digital watermarking, video compression, and image and signal processing.

Dr. Alattar is a member of the SPIE Society.

Eugene T. Lin (S'99) was born in Stillwater, OK, in 1973. He received the B.S. degree in computer and electrical engineering in 1994 and the M.S. degree in electrical engineering in 1996, both from Purdue University, West Lafayette, IN, where he is currently working toward the Ph.D. degree in video watermarking techniques.

He was an intern at Lucent Technologies during the summer of 2000. In 2001 and 2002, he was a summer intern at Digimarc Corporation. His research interests include video watermarking and steganography, as well as video coding and image processing.

Mr. Lin is a member of Eta Kappa Nu.

Mehmet Utku Celik (S'98) received the B.Sc. degree in electrical and electronic engineering in 1999 from Bilkent University, Ankara, Turkey, and the M.Sc. degree in electrical and computer engineering in 2001 from the University of Rochester, Rochester, NY, where he is currently working toward the Ph.D. degree.

In 2001 and 2002, he was a summer intern with Digimarc Corporation. Currently, he is a Research Assistant in the Electrical and Computer Engineering Department, University of Rochester. His research interests include digital watermarking and data hiding—with emphasis on multimedia authentication—image and video processing, and cryptography.

Mr. Celik is a member of the ACM and the IEEE Signal Processing Society.