

# Evaluation of Watermarking Low Bit-rate MPEG-4 Bit Streams

Adnan M. Alattar<sup>a</sup>, Mehmet U. Celik<sup>b</sup>, Eugene T. Lin<sup>c</sup>

<sup>a</sup>Digimarc Corporation, Tualatin, OR 97062

<sup>b</sup>University of Rochester, Rochester, NY 14627-0126

<sup>c</sup>Purdue University, West Lafayette, IN 47907

## ABSTRACT

A novel watermarking algorithm for watermarking low bit-rate MPEG-4 compressed video is developed and evaluated in this paper. Spatial spread spectrum is used to invisibly embed the watermark into the host video. A master synchronization template is also used to combat geometrical distortion such as cropping, scaling, and rotation. The same master synchronization template is used for watermarking all video objects (VOP) in the bit-stream, but each object can be watermarked with a unique payload. A gain control algorithm is used to adjust the local gain of the watermark, in order to maximize watermark robustness and minimize the impact on the quality of the video. A spatial and temporal drift compensator is used to eliminate watermark self-interference and the drift in quality due to AC/DC prediction in I-VOPs and motion compensation in P- and B-VOPs, respectively. Finally, a bit-rate controller is used to maintain the data-rate at an acceptable level after embedding the watermark. The developed watermarking algorithm is tested using several bit-streams at bit-rates ranging from 128-750 Kbit/s. The visibility and the robustness of the watermark after decompression, rotation, scaling, sharpening, noise reduction, and trans-coding are evaluated.

Keywords: Video Watermarking, MPEG-4, Synchronization Template, Spread Spectrum, HVS masking

## 1. INTRODUCTION

Despite being an economic opportunity, video streaming is also a major concern for content owners. The Internet and other digital networks are means for freely and widely distributing high fidelity duplicates of digital media, which is a boon for authorized content distribution but also an avenue for major economic loss arising from illicit distribution and piracy. An effective digital-rights-management (DRM) system would allow content providers to track, monitor, and enforce usage rights of their contents in both digital and analog form. It can also link the content to the providers and may promote sales.

Watermarking [1, 2, 3, 4] and encryption are two oft-mentioned techniques proposed for use in digital-rights-management 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. A watermark, however, 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 may even persist in the video when it has been converted from digital to analog form. Video watermarking applications also include tracking video content, broadcast monitoring, and linking the video to its owner to facilitate value-added services that benefit both the owner and the consumers.

The classical approach to watermarking a compressed video stream is to decompress the video, embed the watermark using a spatial-domain or transform-domain watermarking technique, and then recompress the watermarked video. There are three major disadvantages to using the 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 recompressing 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 bit-stream for watermarking (such as encoded DCT coefficients.) Then, the partially decoded bit-

stream 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 bit-stream will produce a syntactically valid bit-stream 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 bit-stream, such as prediction and quantization parameters, and can adjust the watermark embedding accordingly to improve robustness, capacity, and visual quality. This approach also allows a watermark to be embedded without resorting to the computationally expensive motion estimation process during recompression when temporal prediction (motion compensation) is used.

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. 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 DCT coefficients. The technique includes a method for drift compensation. Data rate control is performed by watermarking only non-zero DCT coefficients, and only if the data rate will decrease 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 low data-rate video (<1Mb/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. Our drift compensation method supports the prediction modes in MPEG-4, including spatial (intra-DC and AC) prediction. Watermark detection is performed in the spatial domain, with the use of 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.

## 2. OVERVIEW OF MPEG-4

MPEG-4 is an object-based standard for coding multimedia at low bit-rates [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. Video compression field-tests at rates below 1 Mbit/s have consistently indicated better performance for MPEG-4 than for MPEG-1 and 2.

The discussion of this paper will be limited to the watermarking of the natural video objects. A natural Video Object (VO) in MPEG-4 may correspond to the entire scene or a physical object in the scene. A VO is expected to have a semantic meaning such as car, tree, man, and etc. 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 to be displayed. Video objects may have arbitrary shapes. The shape information is encoded using 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 2. This algorithm uses motion compensation to reduce inter-frame redundancy and the DCT to compact the energy in every 8x8 block of the image into a few coefficients. It, then, adaptively quantizes the DCT coefficients in order to achieve the desired low bit-rate. It also uses Huffman codes to encode the quantized DCT coefficients, the motion vectors, and most control parameters in order to reduce the statistical redundancies in the data. All coded information is assembled into an elementary bit-stream that represents a single video object. MPEG-4 has enhanced coding efficiency that can be partially attributed to sophisticated DC coefficient, AC coefficient, and motion vector prediction algorithms, as well as, the use of Overlapped Block Motion Compensation (OBMC).

In order to enable the use of 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. Due to its simplicity, improved video quality and compression efficiency, the advanced simple profile is being promoted by the Internet Streaming Media Alliance (ISMA) for delivering video over the Internet.

### 3. WATERMARK STRUCTURE

In the proposed method, a watermark signal is inserted directly into the MPEG-4 compressed bit-stream while detection is performed using the uncompressed video. This allows watermark detection even if video has been manipulated or converted to another format.

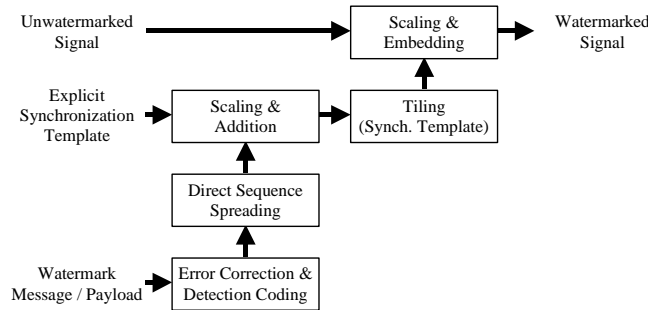


Figure 1. Formation of elementary watermark signal.

Figure 1 depicts the construction of our elementary watermark signal, which comprises of a spread spectrum message signal and a pair of synchronization templates. Spread-spectrum techniques provide reliable data transmission even in very low signal-to-noise ratio (SNR) conditions. A 31-bit payload is used for watermarking each MPEG-4 video object. Error correction and detection bits are 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.

A pair of templates is imposed on the spread spectrum signal to combat synchronization loss due to rotation and scaling of the video after watermark embedding. The watermark detector can examine the templates to determine the orientation and scale of the watermark, and then reverse any such modifications 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 non-overlapping fashion. This tiled structure of the watermark can be easily detected by autocorrelation [9, pp. 271-273]. If the watermark tile is appropriately designed, a peak occurs at the center of each tile. When a pseudo-random 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.

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  $m(x,y)$ . 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 pseudo random phase. The random phase makes the signal look somewhat random in the spatial domain. Since the magnitude of the 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. 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, which can be detected using a Phase Only Match filter (POM).

### 4. WATERMARKING MPEG-4 COMPRESSED DOMAIN VIDEO

This section describes embedding the watermark directly to the bit-stream generated in accordance with the Advanced Simple Profile (ASP) of the MPEG-4 standard. 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, it does not support arbitrary-shaped objects, scalability, interlaced video, and sprites. Note that the techniques and methodology employed in this section can be extended to other profiles of the standard, often with only minor modifications.

## 4.1. Watermark Embedding

At the system level, the watermark embedder mimics the system decoder model described by MPEG-4 standard [8]. The Delivery layer extracts access units (SL 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 bit-streams. Finally, the elementary bit-streams are parsed and watermarked according to the scene description information. The Sync layer re-packetizes the elementary bit-streams 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.

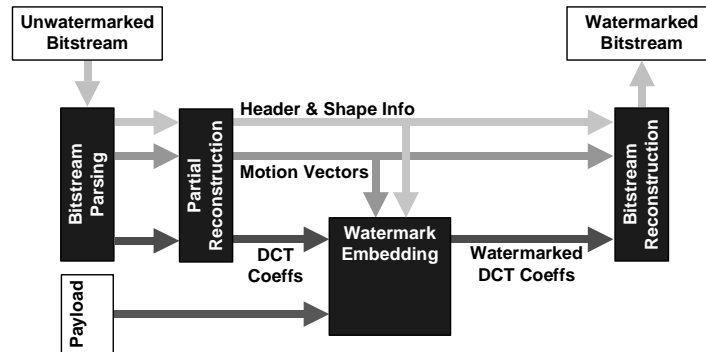


Figure 2. The basic steps of the compressed domain watermarking method.

An elementary bit-stream is parsed down to the block level and variable length coded motion vector and DCT coefficients are obtained as shown in Figure 2. The watermark  $w(x,y)$  is added to the spatial representation of the luminance plane of the VOPs as shown in Figure 3. Before adding the watermark  $w(x,y)$  to the VOPs,  $w(x,y)$  is divided into 8x8 non-overlapping blocks, transformed to the DCT domain, and the DC coefficient of each block is set to zero. The latter step is necessary in order to maintain the integrity of the bit-stream by preserving the original direction of the DC prediction. Removal of 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 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 (CBP) to properly match the coded and not coded blocks after watermarking.

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

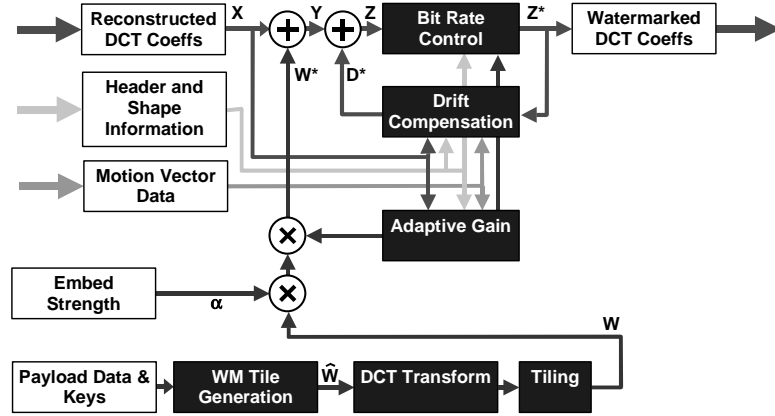


Figure 3. Watermark generation and insertion to the reconstructed DCT coefficients.

#### 4.2. Adaptive Gain (Local Gain Control)

Our local gain control method is applicable for both intra-coded VOPs as well as predicted VOPs. It uses a local activity measure to adjust the watermark power on a block-by-block basis according to the following equation:

$$W_i^*(x, y) = \alpha L_i W_i(x, y) \quad (1)$$

where  $W_i^*$  is the watermark that will be embedded in  $i^{\text{th}}$  MPEG-4 block (8x8 pixels in size),  $\alpha$  is the user-selected global gain,  $W_i$  is the watermark signal prior to gain adjustment.  $L_i$  is given by

$$L_i = \frac{\sqrt{A_i}}{\frac{1}{M} \sum_{k=1}^M \sqrt{A_k}} = \frac{\sqrt{A_i}}{\text{mean}(\sqrt{A_k})} \quad (2)$$

where  $A_i$  is an activity measure. For all intra-coded blocks,

$$A_i = \frac{\text{end}}{k=\text{start}} [\text{DCT}_k]^2 \quad (3)$$

where  $\text{DCT}_k$  is the reconstructed value of the  $k$ -th DCT coefficient in zig-zag order ( $\text{DCT}_0$  is the DC coefficient). For all predicted blocks, equation (3) is not an appropriate activity estimator because the encoded DCT values in the bit-stream represent the prediction residual from motion compensation and not the base-band image itself. Therefore, we adopt another method which uses motion vector information to estimate the activity of predicted blocks in a manner similar to motion compensation. In this case,  $A_i$  is a weighted average of the four reference blocks that has overlap with the current block after applying motion compensation. Specifically,  $A_i$  is estimated by

$$A_i = \frac{\text{RN}_A}{\text{TM}N} \left\} A_A + \frac{\text{RN}_B}{\text{TM}N} \left\} A_B + \frac{\text{RN}_C}{\text{TM}N} \left\} A_C + \frac{\text{RN}_D}{\text{TM}N} \left\} A_D \quad (4)$$

where  $A_A$ ,  $A_B$ ,  $A_C$ , and  $A_D$  are the activity estimates of the four reference blocks,  $N_A$ ,  $N_B$ ,  $N_C$ , and  $N_D$  are number of pixels in the overlap areas between the current block after motion compensation and the four reference blocks, respectively, and  $N$  is number of pixels in the current block.

#### 4.3. Drift Signal Compensation

A spatial domain drift compensator similar to that of [5] is employed to cope with unwanted interference and visual degradations due to motion compensation. In particular, the drift compensator keeps track of the difference between the un-watermarked 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 of the encoder and the decoder, even in the presence of quantization noise. A block diagram for the

drift compensator is shown in. Note that the motion compensation may be performed directly in DCT domain as described in [5], eliminating the transform overhead and reducing computational requirements.

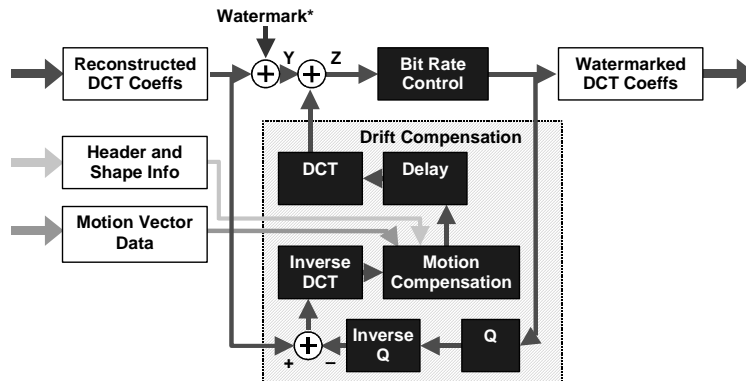


Figure 4. Drift between the encoder and decoder due to watermarking is compensated by a feedback loop

#### 4.4. Bit-Rate Control

Often, watermarked video requires substantially more bits to encode (compress) than un-watermarked video. 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 bit-rate constraints. Herein, we take an alternative approach: First the watermark signal is added to a block of the VOP, and then the quantized DCT coefficients are selectively eliminated (set to zero) until the target bit-rate for the block is met. In each turn, the DCT coefficient with the minimum absolute value is selected. This decreases the number of non-zero DCT coefficients, thus the number of bits required. Note that our algorithm does not differentiate between the host signal and the watermark when eliminating coefficients. As a result, in some instances, host signal quality is sacrificed instead of reducing the amount of embedded watermark. This functionality is especially useful for 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 [10]. Herein, the problem is revisited in the watermarking context, where the fidelity of the embedded watermark signal is traded with that of the host signal through bit allocation. Hereunder, we introduce two heuristic approaches and defer the theoretical optimization problem for future research. First method is a simple strategy that piggybacks to the encoder's bit-rate control algorithm. In particular, bit-budget of a block ( $B_{new}$ ) is determined by the number of original bits ( $B_{orig}$ ).

$$B_{new} = B_{orig} (1 + R) + \Delta \quad (5)$$

where  $R$  denotes the permitted rate of increase, and  $\Delta$  is the number of bits that have been assigned to a previous block but have not been used. Often, bit-rate controller of the encoder, e.g. TM5, allocates more bits to textured areas of the VOP [10]. As a result, more bits are allocated for the watermark in those areas, allowing for a more accurate representation of the watermark. This behavior is in agreement with the local gain adaptation algorithm, which increases the gain in textured areas.

Our second method explicitly uses the local watermark strength and allows for a more flexible approach. Bits available for watermarking are allocated (*Allocation*) among blocks within a VOP according to their local watermark gain factor ( $G_{local}$ ). In particular, at each step the remaining additional bits are shared among remaining blocks according to their local gain factor. In addition, the desired increase (*Increase*), which is the difference between the original number of bits and those required by the watermarked block before coefficient elimination, are taken into consideration. In particular, bit-budget of a block is computed as the original budget plus the average of the allocated and desired increases.

$$\begin{aligned}
B_{new} &= B_{orig} + \frac{\text{TM} \cdot \text{Increase} + \text{Allocation}}{2} \\
\text{Increase} &= B_{watermarked} - B_{orig} \\
\text{Allocation} &= \frac{G_{local}}{G_{local}} \cdot B_{orig} * R
\end{aligned} \tag{6}$$

This method allows for occasional local increases in the budget, and has provided better overall visual quality and/or better watermark robustness in our experiments.

#### 4.5. Watermark Detection

Since a spatial watermark was used, watermark detection is performed after decompressing the bit-stream. 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.

### 5. IMPLEMENTATION AND RESULTS

#### 5.1. Test Setup

Our algorithm has been tested with the first five seconds of the standard sequences: *Foreman*, *Flower Garden*, *Football*, and *Salesman*. All sequences were encoded with MPEG-4 at 128 Kb/s (QCIF 176x144), 384 Kb/s and 768 Kb/s (CIF 352x288) at 15 frames/sec. Resulting bit-streams are supported under ASP and selected bit-rates are in accordance with ASP levels L0 to L3. The sequences are 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 has been computed.  $\Delta$ PSNR value, which signifies the ratio of distortions due to compression and watermarking, [5] has been 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 has been 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 second period). “Per-second detection rate” is meaningful for applications which require at least one detection within a given interval. It also differentiates between bursts of detections versus consistent detections.

Robustness of the algorithm is tested in five categories: decompression only (no attack), filtering, scaling, rotation, and trans-coding. Filtering operations include 3x3 Gaussian and unsharp masking (Matlab default parameters), and Gamma correction ( $\gamma = 0.8$ ). Scaling operations include scaling in spatial dimensions with factors of 75%, 90%, 110%, and 125%, and rotation is performed for 1°, 3°, and 5° (with bilinear sampling). In transcoding<sup>1</sup>, bit-streams have been decompressed and re-compressed at the same bit-rate using a different GOV structure (I-VOP followed by 19 P-VOPs).

---

<sup>1</sup> Limited capabilities of the available MPEG-4 encoder prevented us from transcoding the 768 kb/s sequences.

## 5.2. Experimental Results

Sequence	Unmarked		Marked			Detection Percentage ( <i>per frame and per second</i> )									
	PSNR (dB)	$\Delta$ Rate (%)	PSNR (dB)	$\Delta$ PSNR (dB)	No Attack		Filtering		Scaling		Rotation		Transcoding		
					/frame	/sec	/frame	/sec	/frame	/sec	/frame	/sec	/frame	/sec	
FlowerG	27.9	5.8	26.4	-1.5	11.3	80.3	10.2	78.8	8.7	76.7	8.7	77.2	5.5	70.5	
Football	29.5	0.7	28.5	-1.0	16.2	75.7	14.2	73.3	12.0	68.5	11.5	62.5	6.3	44.3	
Foreman	35.8	6.2	34.0	-1.7	26.3	96.2	23.8	90.0	18.5	81.8	16.8	86.3	8.5	64.0	
Salesman	38.1	1.5	36.0	-2.1	68.7	98.7	61.2	93.5	52.8	87.2	58.3	96.0	69.3	97.5	
Overall	32.8	3.5	31.2	-1.6	30.6	87.7	27.3	83.9	23.0	78.5	23.8	80.5	22.4	69.1	

Table 1. Average quality and increase in bit-rate of each test sequence and the corresponding detection rate

Bit Rate (Kb/s)	Unmarked		Marked			Detection Percentage ( <i>per frame and per second</i> )									
	PSNR (dB)	$\Delta$ Rate (%)	PSNR (dB)	$\Delta$ PSNR (dB)	No Attack		Filtering		Scaling		Rotation		Transcoding		
					/frame	/sec	/frame	/sec	/frame	/sec	/frame	/sec	/frame	/sec	
128.0	31.4	2.6	29.7	-1.7	17.5	81.1	16.0	80.1	13.9	65.8	14.4	68.5	17.5	75.4	
384.0	31.9	1.4	30.8	-1.2	30.5	82.0	26.8	78.6	21.8	76.4	23.4	77.9	27.3	62.8	
768.0	35.1	6.6	33.3	-1.8	43.9	100.0	39.3	93.0	33.4	93.5	33.8	95.1			

Table 2. Average quality and increase in bit-rate for each of the tested bit-rates and the corresponding detection rate

All test sequences have been watermarked using the proposed method and two different global embedding strengths, which have been empirically determined. Local gain control, drift-compensation and bit-rate control algorithms have been turned on and a 10% increase in the bit-rate has been allowed. The *start* and *end* values used in the adaptive gain activity estimation (equation (3)) were 10 and 63, respectively, with the start value chosen empirically to prevent strong edges (which have low-frequency components) from influencing the activity estimation too greatly.

Table 1 and Table 2 show the performance of the technique for all “attacks”. On average, watermarking process has increased size of the compressed bit-stream by 3.5%, whereas the PSNR of the compressed sequence has been decreased by 1.6 dB. It was observed that this amount of degradation is visually more tolerable 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 as “objectionable” at 128 Kb/s. (These observations are further validated by the subjective tests, see Sec. 5.2.1.) This can be attributed to the fact that at lower data rates, the compressed bit-stream 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, watermark is correctly decoded, on average, from more than 30% of the frames with no attack and more than 20% of the frames under various manipulations. In a given one second interval (15 frames), these detection rates translate to a success rate of approximately 90% and 80%, respectively.

In general, higher detection rates are obtained at 768 Kb/s and 384 Kb/s than at 128 Kb/s. Moreover, CIF video detects better than QCIF video, because CIF images provide more data to the averaging processes used to calculate the sync signal. It was also observed that the watermark detection rates are 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.

### 5.2.1. Subjective Quality Test Results

In order to assess the visual effects of the watermarking method subjectively, we ran an informal test with a small number of subjects. In this non-blind test, nine (9) subjects were shown the original and three watermarked (with different embedding strengths) versions of each sequence and asked to “rate the distortion they perceived” as according to the following scale: *Not Noticeable*=5, *Almost Noticeable*=4, *Acceptable*=3, *Somewhat Objectionable*=2, and *Objectionable*=1. The responses are gathered and the Mean response from all subjects for the two embedding strengths, for which the detection results have been reported, is seen in Table 3.

Rate (Kb/s)	Sequence				
	<i>Flower Garden</i>	<i>Football</i>	<i>Foreman</i>	<i>Salesman</i>	Avg.
128	2.3	1.8	3.15	2.85	2.5
384	2.6	2.5	3.7	3.75	3.15
768	3	3.45	4.25	4.25	3.75
Avg.	2.6	2.6	3.7	3.6	

Table 3. Average subjective test scores of each watermarked bit-stream and bit-rate

Subjective quality results first validate the fundamental trade-off between the increase in quality distortion and increased watermark strength, and thus improved detection performance. Upon inspection of different bit-rates, it is seen that the subjects find the watermark more objectionable whenever the quality of the underlying compressed bit-stream is lowered. That is, it is more challenging to insert imperceptible/un-objectionable 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*, are regarded as more acceptable. This may be attributed to the higher quality of un-watermarked 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 is 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, the 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 which normally mask the watermark signal.

### 5.2.2. Performance Improvement with Local Gain Control

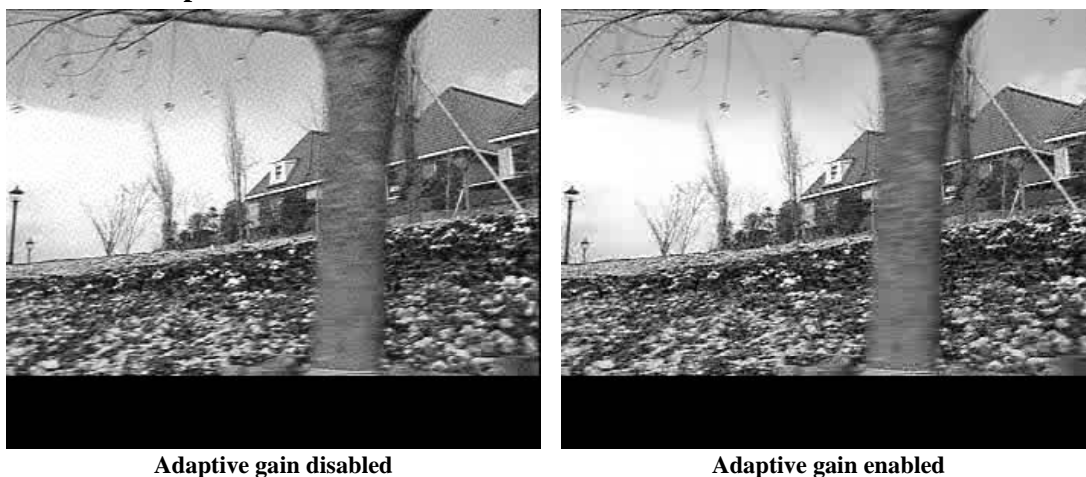


Figure 5. A VOP of watermarked flower garden sequence (384 kbps CIF) with adaptive gain disabled and enabled

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 (MSE) and PSNR may not account for the human perceptual sensitivity to the distortion between two images or videos [11]. Figure 5 shows watermarked VOPs from the *flower garden* sequence with adaptive gain

turned off and on. 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 is much less visible when the adaptive gain is enabled at the same PSNR, however even after examining [11, 12, 13, 14, 15, 16], it was difficult finding an objective measure that consistently shows 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 [15] 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 more perceptually similar. The Universal metric indicated decreasing perceptual quality as global watermark embedding strength is 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 is 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 Figure 6. Three sets of results are shown: Adaptive gain disabled (constant gain), adaptive gain using full-frame reconstruction and equation (3) for all blocks in all VOPs (adaptive gain- exact), and adaptive gain using temporal prediction for activity estimation in predicted blocks as described in Sec.4.1 (adaptive gain-prediction).

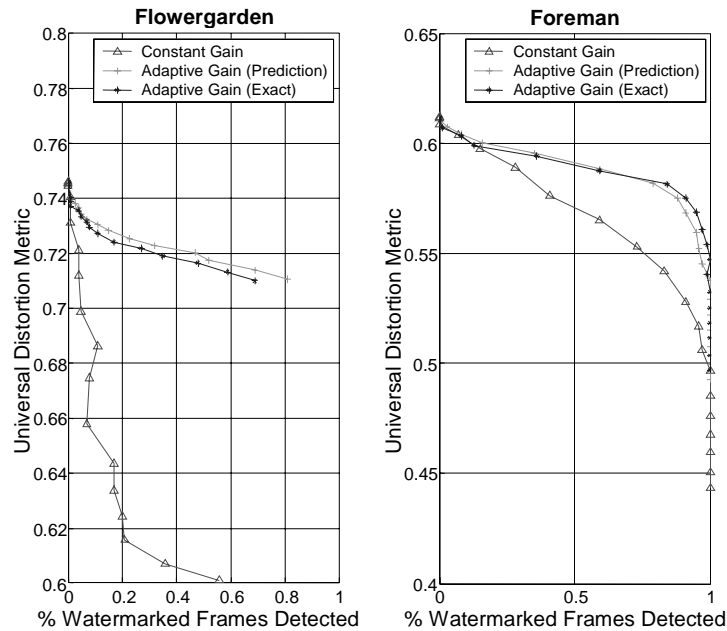


Figure 6. Adaptive gain performance for flower garden and foreman sequences

Figure 6 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 significantly degrade 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 is used. In the existence of motion, the activity estimate from temporal prediction degrades gradually. At an I-VOP, the activity estimate is suddenly corrected using the calculation in (3). 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.

Sequence		Bit-rate limit +0%			Bit-rate limit +10%		
		PSNR (dB)	%Detect /Frame	%Detect /Sec	PSNR (dB)	%Detect /Frame	%Detect /Sec
Flower Garden	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
	Avg.	26.26	5.7	63.0	26.60	7.0	68.3
Football	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
	Avg.	28.65	5.0	43.7	28.70	6.0	51.3
Foreman	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
	Avg.	34.35	7.0	84.0	34.50	18.0	92.3
Salesman	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
	Avg.	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

Table 4. Visual quality and the detection rates with “no attack” with 0% and 10% increase in bit-rate due to watermark

### 5.2.3. Effects of Limited Bit-Rate and Bit-Rate Control

As stated earlier, size of the watermarked bit-stream poses another limitation for compressed domain watermarking. As the bit-rate 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 4). 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.

### 5.2.4. Frame Accumulation for Robust Detection

Sequence	Detection percentage (%)							
	N = 1		N = 3		N = 5		N = 15	
	/Detect	/Sec	/Detect	/Sec	/Detect	/Sec	/Detect	/Sec
Flower Garden	11.3	80.3	33.7	81.8	50.0	85.2	92.5	90.8
Football	16.2	75.7	29.5	85.8	36.2	79.0	68.0	63.3
Foreman	26.3	96.2	46.2	89.8	49.5	77.8	65.2	66.2
Salesman	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

Table 5. Average detection rates when  $N$  frames are averaged before detection

In the experiments presented so far, watermark detection has been performed on each individual frame (VOP). Since the same watermark is inserted in each frame in a GOV, watermark signal contains a significant temporal redundancy, which may be exploited for improved detection performance. Here, frames within a sliding window of size  $N$  are averaged and the average frame is used for detection.

In our experiments,  $N$  is set to 1, 3, 5, and 15. In Table 5, we observe that the success rate for each detection increases significantly through this method (from 30.6% to 76.9%). Nevertheless, the percentage of one second intervals where a watermark is successfully detected (per second detection rate) does not necessarily improve. Upon close inspection of results, it has been observed that often —especially for small  $N$ — a single frame within the sliding window forces a detection. As all window positions that include said frame are successfully detected, the success rate increases 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 get the first detection. Since it is often sufficient to obtain a

single detection, this significantly reduces the computational requirements of the watermark detector. In exchange, a buffer of size  $N$  frames is required.

## 6. CONCLUSIONS

A technique for watermarking MPEG-4 low-bit rate compressed bit-streams was developed and implemented. The technique requires bit-stream parsing and partial decoding, but avoids full decoding and re-encoding of the bit-stream, 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, a novel method for controlling the data rate of the watermarked video that is suitable for low bit-rate video, and a drift compensator that 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. 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/second frame-by-frame detection rate. The extension of our watermarking technique to other MPEG-4 profiles can be achieved often with minor modification.

## REFERENCES

- 
- [1] G. Langelaar, I. Setyawan, and R. Lagendijk, "Watermarking digital image and video data: A state-of-the-art overview," *IEEE Signal Processing Magazine*, vol. 17, no. 5, pp. 20-46, September 2000.
  - [2] M. Swanson, M. Kobayashi, and A. Tewfik, "Multimedia data-embedding and watermarking technologies," *Proceedings of the IEEE*, vol. 86, no. 6, pp. 1064-1087, June 1998.
  - [3] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1079-1107, July 1999.
  - [4] I. Cox, M. Miller, J. Bloom, *Digital Watermarking*, San Francisco: 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] International Organization for Standardization, ISO/IEC 13818-2, *Information Technology – Generic coding of moving pictures and associated audio information*, 1994.
  - [8] International Organization for Standardization, ISO/IEC 14496-2, *Information Technology—Coding of Audio-Visual Objects: Video*, October 1998.
  - [9] S. Haykin, *Communication Systems*, 3rd ed., John Wiley and Sons.
  - [10] G. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, November 1998, pp. 74-90.
  - [11] A. Basso, I. Dalgic, F. Tobagi, and C. Lambrecht, "Study of MPEG-2 coding performance based on a perceptual quality metric," *Proceedings of the 1996 Picture Coding Symposium*, Australia, March 1996, pp. 263-268.
  - [12] A. Webster, C. Jones, M. Pinson, S. Voran, S. Wolf, "An objective video quality assessment system based on human perception," *Proceedings of the Human Vision, Visual Processing, and Digital Displays IV*, Feb. 1993, San Jose, CA, pp. 15-26.
  - [13] S. Westen, R. Lagendijk, J. Biemond, "Spatio-temporal model of human vision for digital video compression," *Proceedings of the SPIE Human Vision and Electronic Imaging II*, Rogowitz, Pappas, Editors, vol. 3016, San Jose, CA, 1997, pp. 260-268.
  - [14] A. Watson, J. Hu, J. McGowan III, "Digital video quality metric based on human vision," *Journal of Electronic Imaging*, vol. 10, no. 1, pp. 20-29, 2001.
  - [15] Z. Wang and A. Bovik, "A universal image quality index," *IEEE Signal Processing Letters*, vol. 9, no. 3, March 2002.
  - [16] Weighted PSNR for images as in Certimark: S. Voloshynovskiy, S. Pereira, V. Iquise and T. Pun, "Attack modeling: Towards a second generation benchmark," *Signal Processing, Special Issue: Information Theoretic Issues in Digital Watermarking*, May, 2001. V. Cappellini, M. Barni, F. Bartolini, Eds.