Watermarking Low Bit-rate Advanced Simple Profile MPEG-4 Bitstreams

Adnan M. Alattar
Digimarc Corporation
19801 SW 72nd Ave., Suite 100
Tualatin, OR 97062
aalattar@digimarc.com

Eugene T. Lin*
Purdue University
School of Electrical and Computer Eng.
West Lafayette, IN 47907
linet@ecn.purdue.edu

Mehmet U. Celik*
University of Rochester
Electrical and Computer Eng. Dept.
Rochester, NY 14627-0126
celik@ece.rochester.edu

ABSTRACT

This paper presents a novel watermarking method for low bit-rate video that is compressed according to the Advanced Simple Profile of MPEG-4. A spatial spread spectrum watermark was embedded directly to the MPEG-4 bit-streams by adopting Hartung's approach of watermarking MPEG-2 compressed bit-streams. A synchronization template was employed to combat cropping, scaling, and rotation. A gain control algorithm adjusts the local strength of the watermark depending on local image characteristics, in order to maximize watermark robustness and to minimize the impact on the quality of the video. A drift compensator prevents the accumulation of watermark distortion and reduces inter-frame interference of watermark signals due to motion compensated prediction in inter-coded frames. The developed watermarking algorithm was tested at bit-rates ranging from 128-768 Kbit/s. The watermark's impact on visual quality as well as its robustness after decompression, scaling, rotation, sharpening, and noise reduction was evaluated.

1. INTRODUCTION

MPEG-4 is an object-based standard for coding multimedia at low bit-rates [1]. Video compression field-tests at rates below 1 Mbit/s have consistently indicated better performance for MPEG-4 than for MPEG-1 and 2. Increased compression efficiency and flexibility of the standard prompted Internet Streaming Media Alliance (ISMA) to promote Advanced Simple Profile (ASP) of MPEG-4 for broadband Internet multimedia streaming. Using MPEG-4 for delivering video on the Internet is gaining momentum and is expected to be ubiquitous.

Despite being an economic opportunity, video streaming is also a major concern for content providers. The Internet can be a means for freely and widely distributing high fidelity duplicates of digital media, and represents a major potential loss of revenue to the content providers. 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 will also link their contents to the providers in order to promote more sales. MPEG-4 Intellectual Property Management and Protection (IPMP) extension standardizes a generic interface to proprietary IPMP tools and DRM systems.

Although encryption plays an important role in DRM, it can only protect the digital content during transmission from the content provider to the authorized user. Once the content is decrypted or converted to analog form, encryption fails to provide any protection. On the other hand, watermarking technology can be used to provide the necessary protection after the content is decrypted and even after it is converted to analog form. Watermarking can be interfaced to MPEG-4 through IPMP hooks, potentially preventing playback when the content is captured in analog form and recompressed. It also can be used to link content to its provider, allowing for value-added services.

Several researchers evaluated watermarking in the context of MPEG-4 compression. The authors of [2]-[4] investigated the watermarking of individual video objects in the spatial domain. Nicholson et al. [5] evaluated watermark robustness and video quality after the video is watermarked and compressed by MPEG-4 standard at bit-rates ranging from 0.250 to 8 Mbit/s.

It is faster and more flexible, however, to apply the watermark directly to the MPEG-4 compressed bit-stream. This allows the embedder to shape the watermark according to the compression parameters in order to optimize robustness and quality. It also allows content providers to customize the watermark on demand without resorting to the complex and time-consuming process of decompression, watermarking, and then recompression.

Hartung has developed an algorithm to watermark MPEG-2 bit-streams without completely decompressing them [6]. Hartung's approach is based on extracting the discrete cosine transforms (DCT) coefficients from the bit-stream, then adding the watermark to the extracted DCT coefficients. Hartung evaluated his approach with compressed bit-stream at rates that vary between 4 and 12 Mb/s. These rates are more suitable for DVD and digital TV broadcast video than for low bit-rate Internet video (<1Mb/s).

Herein, Hartung's approach is applied to watermark MPEG-4 advanced simple profile bit-streams. Our watermark signal is inserted in the MPEG-4 compressed domain and is robust against geometric distortions and common filtering operations. Although MPEG-4 is similar to MPEG 1-2 in principle, MPEG-4 is object-oriented and includes additional coding features that require some adjustment to Hartung's basic approach to watermarking. Moreover, the nominal low-bit rate of MPEG-4 requires special design for the watermark in order to achieve robustness without increasing the visibility of the watermark.

2. SPREAD-SPECTRUM WATERMARK

Our watermark is a spatial domain pseudo-random noise signal covering the entire video object. Spread spectrum allows for

* E. Lin and M. Celik were interns with Digimarc Corporation, Tualatin, OR.

reliable detection, even when the embedded watermark signal is impaired due to the interference from the host signal and noise arising from subsequent processing. Nevertheless, a spread spectrum watermark is vulnerable to synchronization error, which occurs when the watermarked signal undergoes geometric manipulations such as scaling, cropping and rotation.

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 in the literature. A pair of templates is imposed on the spread spectrum signal to combat synchronization loss. In particular, using the synchronization templates, the change in rotation and scale after watermark embedding is determined and "undone" prior to detection of the message using spread spectrum techniques.

2.1 Synchronization Templates

In our watermark, two synchronization templates are imposed on the watermark signal. The first template 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 *NxM)* in a non-overlapping fashion. This tiled structure of the watermark can be easily detected by autocorrelation, where a peak at the center of each tile occurs if the watermark signal is appropriately designed. If a linear transformation *A* is applied to a watermarked VOP, the autocorrelation coefficients *η(x,y)*, thus the peaks, move to new locations (x',y') according to the following equation

$$\begin{bmatrix} x' & y' \end{bmatrix}^T = A\begin{bmatrix} x & y \end{bmatrix}^T \tag{1}$$

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. 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

$$\begin{bmatrix} u' & v' \end{bmatrix}^T = \left(A^T\right)^{-1}\begin{bmatrix} u & v \end{bmatrix}^T \tag{2}$$

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

$$A = \begin{bmatrix} S\cos\theta & -S\sin\theta \\ S\sin\theta & S\cos\theta \end{bmatrix} \tag{3}$$

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 as the center of the image. Under the log-polar mapping, the transformation *A* becomes [8]:

$$\begin{bmatrix} \log\rho' \\ \alpha' \end{bmatrix} = \begin{bmatrix} \log\rho \\ \alpha \end{bmatrix} + \begin{bmatrix} \log S \\ \theta \end{bmatrix} \tag{4}$$

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).

2.2 Message Signal

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 *NxM* 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 also added to the message to protect it from channel errors caused by the host image or distortion noise added by normal processing or intentional attacker.

In order to reduce visibility and the effect of the host image on the watermark, spread spectrum modulation was used with the message bits. First, the values 0,1 were mapped to –1 and 1, respectively. Then, each bit was multiplied by a different pseudo random code of length *K* producing a spread vector of size *K*. Finally, an *NxM* tile was 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 of the watermark signal 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 following:

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

where *a* and *b* are predetermined constants that control relative power between the message and the synchronization signals.

3. WATERMARKING ADVANCED SIMPLE PROFILE

This section discusses embedding the watermark directly to the bit-stream generated in accordance with the Advanced Simple Profile (ASP) of MPEG-4. ASP supports all capabilities of MPEG-4 Simple Profile in addition to B-VOPs (Video Object Planes), quarter-pel motion compensation, extra quantization tables, and global motion compensation. This profile does not support arbitrary-shaped objects, scalability, interlaced video, and sprites.

3.1 Watermark Embedding

The watermark embedder mimics the system decoder model described by MPEG-4 standard [7]. 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.

In order to add the watermark *w(x,y)* to the luminance plane of the VOPs (Video Object Planes), full decompression of the elementary bit-stream is not necessary. Since the DCT is a linear transform, the embedder can add the watermark directly to the DCT coefficients of the luminance blocks. Hence, only the DCT coefficients must be parsed from the elementary bit-stream. All other information will be used to re-assemble the watermarked bit-stream and must be retained.

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 coefficients are 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. 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 ZigZag scan order, reversing the AC prediction (if applicable), and inverse quantization using the given quantizater 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 [6], 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), ZigZag scanning, constructing run-value pairs and VLC coding
7. *Adjust the Coded-Block Pattern (CBP) to properly match the coded and not coded blocks after watermarking.*

The content-adaptive gain mechanism adjusts the amplitude of the watermark signal of each 8x8 block using the variance of the compressed image signal (with increasing variance corresponding to increased gain.) For intra-coded blocks, the variance of each block can be computed directly from the DCT coefficient data. For inter-coded blocks, the variance is estimated by averaging the estimated variance information of the reference VOP using motion vector information (and ignoring the coded residual for the inter-block.)

Once all the blocks in a VOP are processed, the bit-stream corresponding to the VOP is re-assembled. The embedding of the watermark often causes the data rate of the watermarked video to increase. Currently, a bit-rate control algorithm is being investigated that preserves the data rate of the watermarked video (within a specified tolerance) by zeroing the smallest non-zero DCT coefficients in a VOP.

3.2 Watermark Detection

Since a spatial watermark was used, watermark detection is performed after decompressing the bit-stream. Spatial domain implementation has an advantage over compressed domain detection: The watermark can be recovered if the video undergoes decompression and re-compression, or even after it is converted to analog domain, regardless of the recompression parameters or analog format used. The 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.

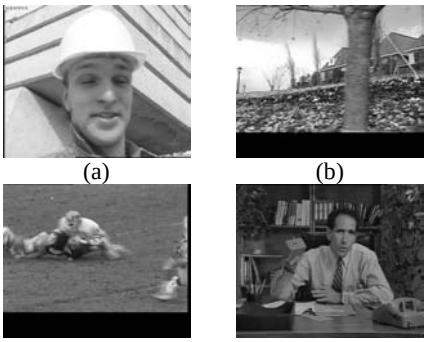
Currently, only the template imposed by the embedding of the synchronization signal *g(x,y)* is utilized for synchronization. The scale and orientation of the VOP is resolved using *g(x,y)* and the log-polar re-maping described in Section (2.1), as follows: First, the VOP is divided into blocks of size *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 zero. This selective processing of the blocks enhances signal-to-noise ratio (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. Then average magnitude of the FFT of all these blocks is computed and used to calculate the re-mapping described in equation (4). Finally, the linear shifts in equation (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.

4. SIMULATION RESULTS

Our algorithm was tested with the first five seconds from the standard sequences: *Foreman*, *Flower Garden*, *Football*, and *Salesman*. These 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, in accordance with ASP profile levels (L0-L3). The GOV structure was comprised of an I-VOP followed by 14 P-VOPs. Then, the compressed bit-streams were watermarked, allowing a 10% maximum increase in the bit-rate. This was done by adjusting the gain of the watermark manually. Figure 1 shows the quality of the first frame of each of the watermarked test sequences at 768 Kb/s. Table 1 lists the average Peak Signal to Noise Ratio (PSNR) values before and after watermarking as well as the watermark detection results.

The average PSNR of the compressed sequence before and after watermarking was calculated with respect to the original uncompressed video. The difference between these PSNR values (ΔPSNR, as in [6]) was the distortion due to watermarking.



(a) (b) (c) (d)

Figure 1 First frame of the test sequences after watermarking
(a) Foreman, (b) Flower Garden, (c) Football, (d) Salesman

As shown in Table 1, the watermarking procedure increases the distortion by approximately 1-2dB. 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. This can be attributed to the fact that at low 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 bit-rates, the watermark can be embedded into visually less significant features. Thus, maintaining video quality, after watermarking, at lower data bit-rates is more challenging.

	Unmarked		Watermarked			Detection Percentage After Manipulation			
	Bit-Rate (Kb/s)	PSNR (dB)	PSNR (dB)	Δ PSNR (dB)	Δ Bit-Rate %	None	Filter-ing	Scale	Rota-tion
Flower Garden	128	27.4	25.5	-1.9	10.1	12	11	9	8
	384	26.8	25.0	-1.8	9.9	23	22	19	17
	768	30.1	28.7	-1.4	9.9	11	9	6	7
	Avg.	28.1	26.4	-1.7	10.0	15.3	14.0	11.3	10.7
Football	128	28.1	26.7	-1.4	0.9	13	9	9	8
	384	28.9	27.7	-1.2	2.4	20	18	16	20
	768	32.1	30.9	-1.2	4.7	31	27	19	19
	Avg.	29.7	28.4	-1.3	2.7	21.3	18.0	14.7	15.7
Foreman	128	34.1	31.9	-2.2	8.9	16	13	10	10
	384	35.4	33.6	-1.8	10.2	32	25	19	16
	768	38.4	36.8	-1.6	9.3	35	30	21	17
	Avg.	35.9	34.1	-1.8	9.5	27.7	22.7	16.7	14.3
Salesman	128	37.0	34.9	-2.1	0.5	32	32	16	22
	384	37.4	36.3	-1.1	-0.1	47	38	24	29
	768	40.3	38.7	-1.6	2.0	52	43	37	47
	Avg.	38.2	36.6	-1.6	0.8	43.7	37.7	25.7	32.7
Overall		33.0	31.4	-1.6	5.7	27.0	23.1	17.1	18.3

Table 1. Quality of the sequence before and after watermarking and the corresponding detection percentage (%).

The robustness of our algorithm was evaluated after decompression, rotation (1°, 3°, 5°), scaling (75%, 90%, 110%, 125%), and filtering (3x3 Gaussian, Un-sharp masking, Gamma correction $\gamma = 0.8$). Table 1 demonstrates the effectiveness of the proposed watermarking algorithm under these manipulations. The results are expressed in terms of the percentage of frames for which the payload is correctly decoded. On average, watermark is correctly decoded from more than 20% of the frames. (Note that the detection was performed independently on each VOP.) Moreover, it was observed that scale changes and rotation have decrease the detection rate somewhat, but the average detection rate is still 17% or 2 detections / sec for 15fps video. In the worst case, detection rate was about 7%, which is equivalent to 1 detection/s (equivalent to embedding 31 bits/s).

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 (see Section (3.2)). 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, resulting in a higher detection rate. In contrary, the watermark in the *Flower Garden* sequence with persistent global motion is harder to detect.

5. CONCLUSIONS & FUTURE WORK

A technique for watermarking MPEG-4 compressed bit-streams was developed and successfully implemented. The watermark can be detected after filtering, scaling, and rotation. The average detection rate can be at least one frame/second with less than 10% increase in the bit-rate. It was observed that watermarking of low bit-rate compressed video signals (<1 Mb/s) is more challenging than watermarking at higher bit-rates. Currently a number of issues and enhancements to our algorithm are being investigated, including bit-rate control, effect of VOP type on performance, trans-coding attack, temporal HVS masking, detection using multiple VOPs, temporal averaging to improve detection, and variable watermark payload within a GOV.

REFERENCES

[1] "Information Technology – Coding of Audio-Visual Objects, Part 2: Visual," *Intern. Standard, ISO/IEC* 1999.

[2] Piva et al., "A DWT-Based Object Watermarking System for MPEG-4 Video Streams," *IEEE-ICIP'00*, Vancouver, Canada, 2000.

[3] M. Barni, F. Bartolini, V. Cappellini, and N. Checca-cci, "Object watermarking for MPEG-4 video streams copyright protection," in *Security and Watermarking of Multimedia Contents II, Wong, Delp, Editors, Proceedings of SPIE Vol. 3671*, San Jose, CA, January 2000.

[4] P. Bas and B. Macq, "A new video-object watermarking scheme robust to object manipulation," *IEEE-ICIP'01*, Thessaloniki, Greece, 2001, pp. 526-529.

[5] D. Nicholson, P. Kudumakis and J.F. Delaigle "Watermarking in the MPEG4 context," *European Conference on Multimedia Applications Services and Techniques*, Madrid, Spain, May 1999, pp.472-492.

[6] F. Hartung, "Digital Watermarking and Fingerprinting of Uncompressed and Compressed Video," *Ph.D Dissertation*, University of Erlangen, 2000.

[7] "Information Technology – Coding of Audio-Visual Objects, Part 1: System," *Intern. Standard, ISO/IEC* 1999.

[8] A. Alattar, J. Meyer, "Watermark re-synchronization using log-polar mapping of image autocorrelation," accepted in the ISCAS'03, Bangkok, Thailand, May 2003.