WATERMARKING AND STREAMING COMPRESSED VIDEO

Oztan Harmanci[1]*, M. Kivanc Mihcak*[2]*, A. Murat Tekalp*[3]

[1]Dept. of Electrical and Computer Engineering, University of Rochester, Rochester, NY
[2]Dept. of Electrical and Electronics Engineering, Bogazici University, Istanbul, Turkey
[3]Dept. of Electrical and Electronics Engineering, Koc University, Istanbul, Turkey

ABSTRACT

In this paper, we propose a novel method for watermarking compressed video for streaming. The proposed method performs motion compensated watermarking. There are two consequences of this; first we see that the impact of the watermark on the final bitrate is negligible, and the proposed method has inherent robustness to mild packet losses in the channel. Experimental results show that the proposed method operates at lower distortion levels and has higher watermark detection rates compared to current watermarking methods. We observe 20% bitrate reduction for the same watermark detection rate. We also show that when data partitioning is used over a packet loss channel the proposed method performs at acceptable watermark detection rates.

Index Terms— video watermarking, video streaming, motion compensated watermarking

1. INTRODUCTION

One application of watermarking is embedding watermark into streaming video. The video is generally stored in a streaming server in compressed format, and upon request watermark is inserted real-time. Generally, complete decoding of the bitstream is avoided, and watermark is inserted into the transform coefficients[1] or the motion vectors[2] as extracted from the compressed bitstream. Since state-of-the-art video codecs are motion compensated hybrid codecs, modifying the residuals affects the future frames and may introduce high levels of distortion if unmonitored. To monitor and control this distortion "drift compensation" is performed. Any watermarking method that works with motion compensated hybrid encoders should perform drift compensation[1]. Drift compensation requires a motion compensation loop in the embedder, which tracks the effect of injected residuals. The complexity is significantly higher than a simple residual coefficient modification. It is shown that uncontrolled residual injection can cause increased bitrates, therefore further rate control, in the form of harsher quantization, may be necessary[3].

Current methods are mostly based on applications of spread spectrum(SS) [4] watermarking to video. Due to special temporal correlations in the video data, the watermark is treated differently in temporal dimension, such that correlations are inserted in the watermark. Temporal decorrelating methods include; temporally static repetitive watermarking[3], temporal transform based methods[5], achieving temporal redundancy by key management [6] or motion compensated watermarking[7].

In this paper we propose a watermarking algorithm for motion compensated hybrid video encoders, based on our previous work[7]. We investigate the encoding performance (i.e. effect of watermark on the final bitrate), and the robustness of the watermark detection under low bitrate compression and packet loss channels. That is, we consider two attacks on the watermarking system; low bitrate compression and packet losses. In the next section we discuss the pseudo random statistics quantization method for watermarking. In Section 3, we discuss the challenges of video watermarking such as the impact of watermark on the bitrate. In Section 4, we discuss the proposed method. Section 5 presents the experimental results and we conclude in Section 6.

2. BACKGROUND

We employ the general approach of watermarking via quantization of pseudo-random (PR) linear statistics [8, 9]. These statistics (also called "hash values") are linear weighted combinations of samples in possibly overlapping regions. Let $\mathbf{s} \in \mathbb{R}^n$ denote the original signal coefficients of size n. There are m statistics denoted by μ_i, $1 \leq i \leq m$. The i-th statistic is represented by the set $R_i \subset \{1, 2, \ldots, n\}$, which defines the indices of the coefficients of $\mathbf{s}$. μ_i of $\mathbf{s}$ is computed by a linear weighted combination of PR weights and $\{s_j\}$ that fall into R_i. We represent the weights for R_i with $\mathbf{t}_i \in \mathbb{R}^n$, where $t_i(j) = 0$ if $j \notin R_i$. Hence, we write $\mu_i = \sum_{j=1}^{n} s(j)t_i(j) =< \mathbf{s}, \mathbf{t}_i >$, which leads to $\mu = \mathbf{Ts}$, where $\mu \in \mathbb{R}^m$ is the vector of statistics, and $\mathbf{T} \in \mathbb{R}^{m \times n}$ is formed such that its i-th row is $\mathbf{t}_i$.

We design the additive WM sequence $\mathbf{w} \in \mathbb{R}^n$ such that the watermarked signal $\mathbf{x} = \mathbf{s} + \mathbf{w}$ has statistics $\hat{\mu} \in \mathbb{R}^m$ that are the quantized version of μ (e.g. a scalar uniform quantizer with step size δ). We compute $\mathbf{w}$ such that $\mathbf{T}(\mathbf{s}+\mathbf{w}) = \hat{\mu}$ and $||\mathbf{w}|| = ||\mathbf{x} - \mathbf{s}||$ is minimized; i.e., we solve

$$\min_{\mathbf{w}} ||\mathbf{w}|| \quad \text{s.t.} \quad \mathbf{Tx} = \hat{\mu} \;\Leftrightarrow\; \mathbf{Tw} = \hat{\mu} - \mu. \tag{1}$$

Assuming that $\mathbf{T}$ is full-rank, which is almost always satisfied with our parameter selection, the solution to (1) is given by the well-known *minimum-norm* (MN) result:

$$\mathbf{w}_{MN} = \mathbf{T}^T \left(\mathbf{TT}^T\right)^{-1} (\hat{\mu} - \mu) \tag{2}$$

To obtain an idea on the behaviour of the WM, note that $\mathbf{t}_i$ are independently generated, therefore

$$\left(\mathbf{TT}^T\right)^{-1} \approx c\mathbf{I} \rightarrow \mathbf{w}_{MN} \approx \mathbf{T}^T (\hat{\mu} - \mu). \tag{3}$$

where c is a scaling factor and $\mathbf{I}$ is the identity matrix.

This equation suggests that weight properties manifest themselves in the final WM, since $\mathbf{w}_{MN}$ is a linear combination of $\mathbf{t}_i$. In [8], we showed that for image watermarking purposes, weights ($\mathbf{t}_i$) should be low-pass band limited for lossy compression and geometric attack robustness. In [7], we showed that for video watermarking purposes, motion compensated weights result in robust WM against temporal estimation attacks and flicker free video.

In the following discussions *encoder* and *decoder* is used for video encoder and video decoder, and *embedder* and *detector* is used for watermark embedder and watermark detector respectively.

3. ISSUES IN WATERMARKING OF COMPRESSED VIDEO

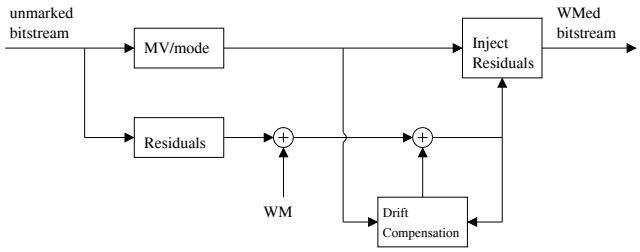

Fig. 1. Flow diagram of a simple additive watermarking method with drift compensation.

Fig. 1 shows a common method of embedding watermark in a pre-coded bitstream. First, the motion vectors, residuals and MB mode information is extracted from the bitstream. The WM is embedded in the residuals additively and the motion vectors and MB modes are generally not changed. Due to motion compensation, any effect on the residuals will propagate in time and spread to other frames. To avoid this, drift compensation is performed by applying motion compensation to the WM embedded in the previous frames.

For the WM to have minimal bitrate impact and have good detection rates, it should also be in accordance with the model assumed by the encoder. In image watermarking literature, we see that robust spread spectrum watermarks remove the high frequency components of the WM. Hence they have higher detection rates when image is compressed. However, in video watermarking literature there is little work utilizing motion compensated prediction done by the video encoder.

The main impact of mismatching host model and WM model arises from the fact the bitstreams are pre-encoded. The transcoding/watermarking operation uses same modes and motion vectors when re-encoding the watermarked video. If, for example, SKIP mode is used for a macroblock, no residuals will be sent for that macroblock, even though the injected residuals and drift compensation signal would require so. This causes a significant performance loss especially at low bitrates, where SKIP mode is widely used. For H.264 the problem is even more severe; commonly used temporally static watermarking methods especially fail in this case because H.264 SKIP modes may have non-zero motion vectors and fail to predict the temporally static watermark.

After WM embedding, the final bitrate may change. Hence state-of-the-art methods employ a rate-control algorithm after WM insertion. Since motion vectors and MB modes are unchanged, the only parameter for rate control is modifying the residuals. In other words, WM is embedded in the residuals, and later residuals are removed for rate control. WE believe this is contradicting since WM detection performance is inherently hindred by design of these algorithms.

To summarize, since motion vectors and MB modes are not changed during watermarking, the bitrate increase is caused by the additional residuals. If we minimize the dependence of the watermark on the residuals, we will minimize the impact of WM on the bitrate. This can be achieved by designing the WM such that it can be predicted by the motion vectors and MB modes *as extracted* from the bitstream. Hence, we propose to have a prediction loop similar to video codecs in the design of the WM, which we will discuss in the next section.

4. PROPOSED METHOD

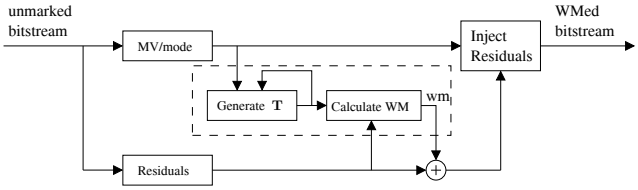

Fig. 2. Flow diagram of the proposed method.

We use the Pseudo-Random Statistics Quantization (PRSQ) watermarking on the luminance samples. In this section we describe a method to implement motion compensated PRSQ watermarking for compressed H.264 video.

4.1. Hash Function Design for Motion Compensated Watermarking

We refer back to Eqn. 3, where we show that the final watermark can be approximated as linear combination of weights ($\mathbf{t}_i$). Therefore, if each of the weights match the host model, the watermark will also match the model. This means; temporally, the weights should be motion compensated, and spatially, they should not contain high frequency components.

Let superscripts denote the temporal location (i.e. frame no); $\mathbf{s}^j$ denotes j-th frame, $\mathbf{w}^j$ denotes the watermark embedded to $\mathbf{s}^j$ and $\mathbf{t}_i^j$ denotes weights used for frame j for the i-th region. Let $\mathbf{l}$ denote a pixel location at $\mathbf{l} = (x, y)$. The motion vector that maps pixel on frame j at location $\mathbf{l}$ to Δ frames past is denoted by $\mathbf{mv}(j, \mathbf{l}, \Delta)$.

The weights in regions R_i that correspond to INTER MBs are generated via motion compensation from weights used at the previous frame(s). This is a recursive process; $\mathbf{t}_i$ is not computed in one step, instead it is computed frame by frame. Each weight is computed independetly from the other weights. The generation is performed as;

$$\mathbf{t}_i^j(\mathbf{l}) = \mathbf{t}_i^{j-\Delta(\mathbf{l})}(\mathbf{l} + \mathbf{mv}(j, \mathbf{l}, \Delta(\mathbf{l}))) \quad (4)$$

where, $\Delta(\mathbf{l})$ denotes the reference frame used for this pixel, and $\mathbf{mv}(j, \mathbf{l}, \Delta(\mathbf{l}))$ is the motion vector. Both $\Delta(\cdot)$ and $\mathbf{mv}(\cdot, \cdot, \cdot)$ are obtained from the pre-coded bitstream. SKIP mode may have zero (MPEG2, MPEG4, etc.) or non-zero (H.264) motion vectors depending on the codec standard. Each weight, $\mathbf{t}_i$, is predicted only from itself, and completely independent from other weights. We use bilinear interpolation to determine weights for sub-pixel motion vectors. It is straightforward to extend the above temporal weight prediction method to biprediction, or weighted multiple reference frame prediction.

Fig. 2 shows the flow diagram of the watermark embedder and the video encoder. When we compare Figures 1 and 2 we see that the main contribution of the proposed method is that, the generated WM relies on modes and motion vectors used in the original compressed video. As a result the WM has minimal impact on the bitrate.

We mentioned that drift compensation is necessary when watermarking is done for motion compensated hybrid encoders. In the

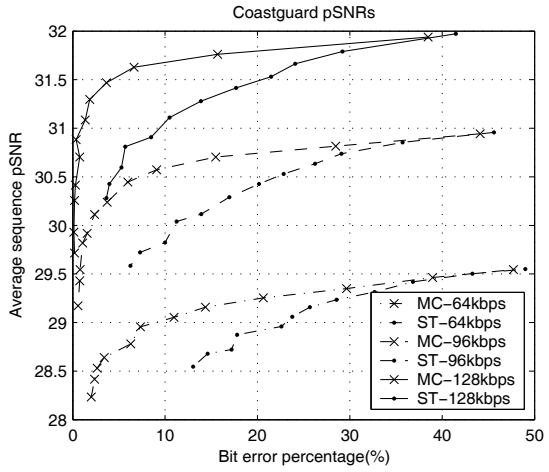

Fig. 3. Achievable BERs at a given distortion level. Distortion is measured with respect to unwatermarked and uncompressed original video. MC represents the proposed motion compensated watermarking method, and ST represents temporally static repetitive watermarking method.

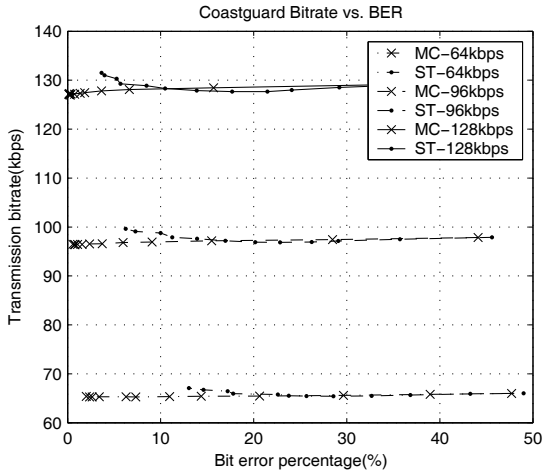

Fig. 4. Achievable BERs for a given bitrate. MC represents the proposed motion compensated watermarking method, and ST represents temporally static repetitive watermarking method.

proposed system this is done by the performing motion compensation in the final encoder shown in Fig. 2. In the proposed method, we will show by experimentation that there is no or negligible increase in the final bitrate, hence a post-embedding rate control method is not necessary.

4.2. Message Embedding

A single bit is embedded in each hash. A simple scalar quantizer is used to quantize each hash to a different lattice depending on the corresponding bit value. Let $\mathbf{M} \in \{0,1\}^m$ denote the message. Then,

$$\hat{\mu}_i = \begin{cases} Q(\mu_i, \delta), & \mathbf{M}_i = 0 \\ Q(\mu_i - \frac{\delta}{2}, \delta) + \frac{\delta}{2}, & \mathbf{M}_i = 1 \end{cases} \quad (5)$$

where $Q()$ is a uniform scalar quantizer and δ is the quantization step size. Actual payload can be less than m bits and forward error correction can be used to recover errors.

4.3. Decoder/Detector Algorithm

Video decoder is a standard compliant decoder, and no extensions are necessary. Watermark detector operates in the spatial domain and uses MV and MB mode information data for each frame as extracted by the decoder from the bitstream.

Let $\tilde{\mathbf{s}}$ denote the received video. The same secret key that is used at the encoder side is used to generate $\tilde{\mathbf{t}}_i$. Note that both the received video and the generated weights can be different than the ones used at the encoder due to channel loss. The detector generates the received hash values; $\tilde{\mu}_i =< \tilde{\mathbf{s}}, \tilde{\mathbf{t}}_i >$. Then the detector determines the quantization lattice and extracts the message. Let $\tilde{\mathbf{M}} \in \{0,1\}^m$ denote the received message, the detection is performed in a maximum likelihood manner as follows;

$$Q\left(\tilde{\mu_i}, \frac{\delta}{2}\right) = b_i \frac{\delta}{2} \rightarrow \tilde{\mathbf{M}}_i = \begin{cases} 0, & b_i \text{ even} \\ 1, & b_i \text{ odd} \end{cases} \quad (6)$$

If forward error correction is used, the detector proceeds with error correction and extracts the message bits.

Watermark detector is designed to be low complexity. This is achieved by the fact that unlike the embedder, the detector can operate frame-by-frame, since it does not need to store the $\mathbf{T}$ matrix. When a new group of pictures start, $\tilde{\mathbf{t}}_i$ and $\tilde{\mu}_i$ are initialized to zero. Then $\tilde{\mathbf{t}}_i^1$ are generated, which are INTRA frame weights and are generated independently. Then weights and hashes are recursively calculated for the following frames. When the decoder decodes frame j, it passes the MV and mode information to the detector and the detector generates the weights corresponding to this frame; $\tilde{\mathbf{t}}_i^j$, $i = 1, ..., m$. Computation of $\tilde{\mathbf{t}}_i^j$ requires using previous weights; $\tilde{\mathbf{t}}_i^{j-k}, k \in \{1, ..., N_{FB}\}$. N_{FB} is the length of the frame buffer used by the encoder/decoder. Since motion compensation cannot exceed N_{FB} frames, we do not need to store weights that correspond to frames further than N_{FB}. Then for $j \geq 2$ each hash is updated;

$$\tilde{\mu}_i^j = \tilde{\mu}_i^{j-1} + < \tilde{\mathbf{s}}^j, \tilde{\mathbf{t}}_i^j > \quad (7)$$

When the necessary amount of frames are received (i.e. the number of frames in a GOP), the detector extracts the message using Eqn. 6. This recursive computation removes the need for extra buffer, which is equal to the size of GOP, for the detector.

5. EXPERIMENTAL RESULTS

We use baseline profile H.264 codec for the experiments. Due to limited paper length, we only present the results for the well known test sequence *Coastguard*, however other sequences results are similar. All sequences are 5 seconds long and encoded at QCIF resolution (176×144) at 15Hz. We use a GOP structure of 1 INTRA frame followed by 14 INTER frames, hence there are 5 GOPs total. Every sequence is encoded for storage first, and then experiments are performed on the encoded bitstreams. Every GOP is embedded 100 bits. We measure the final bit error rate (BER) in terms of total number of bit errors. Note that forward error correction can be applied at both embedder and detector to reliable transmit a shorter message, where the message length depends on the observed BER.

We first study the effect of embedding on the distortion and bitrate. For this, we increase the power of the watermark -which

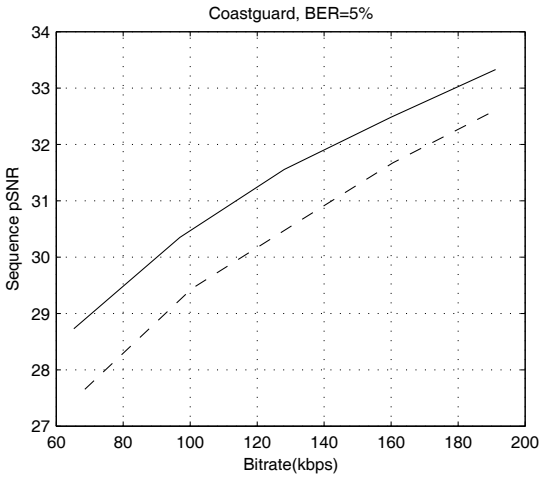

Fig. 5. Operational RD curves for bit error rate 5%. Solid: Proposed, Dashed: Temporal Repetition WM

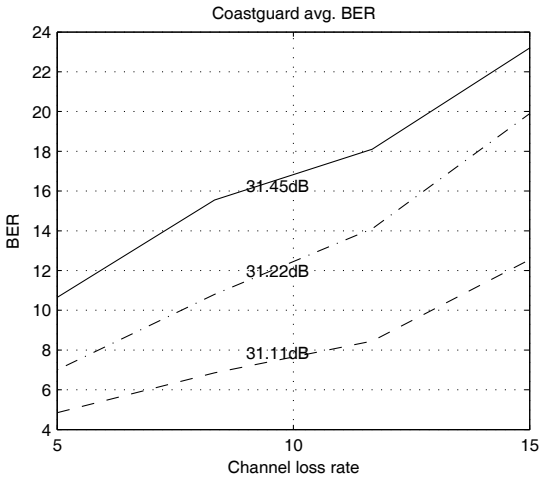

Fig. 6. Watermark detection performance under unequal error protection channel.

is controlled by quantization level δ- and then record the distortions(pSNR) and generated bitrates at the encoder/embedder. We then record the bit error rates observed at the detector. Distortion is measured with respect to the unmarked and uncompressed video.

Fig. 3 compares the distortion and Fig. 4 compares the bitrates of the proposed method vs. the temporally repetitive watermark embedding method for various WM detection BER. Note that, with the proposed method the bitrate increase is negligible, whereas the repetition based methods suffer from rate increase. We combine the two results and generate the RD curves at 5% WM bit error rate in Fig. 5. We see in Fig. 5 that for *Coastguard* sequence, at 5% BER temporally static embedding method requires 170kbps to achieve 32dB, whereas the proposed method requires 142kbps. This corresponds to about 120% more bandwidth. At 100kbps, we see that the proposed method results in about 1dB better quality at the same bitrate to achieve 5% BER.

We also study the performance of the proposed system under unequal error protection channels with data partitioning. The prediction partition is transmitted losslessly and the residual partition is subject to packet loss. Again, watermark power is varied and results are plotted for 3 packet loss rates in Fig. 6 at 128kbps. As expected, increasing the channel loss rate decreases the detection performance. Also, higher power embedding results in better BER rates. We see that at 31.11dB, the proposed method can operate at 12% BER even though the packet loss rate is 15%.

6. CONCLUSIONS

In this paper we propose a method for watermarking compressed video. One application of this is embedding realtime watermarks at a streaming server. The proposed method makes use of encoder's source model, which is motion compensated prediction, to generate motion compensated watermark. As a result, we show that the proposed method requires significantly less distortion -about 1dB- to achieve same watermark detection rates compared to temporally repetitive watermarking, which is widely used by state-of-the-art methods. We also show that the bitrate increase is negligible hence a post-embedding rate control algorithm is not necessary. Finally we also show that the proposed method is robust to mild packet losses in the channel.

7. REFERENCES

[1] F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video," *Signal Processing*, vol. 66, no. 3, pp. 283–301, 1998.

[2] J. Zhang, J. Li, and L. Zhang, "Video watermark technique in motion vector," *Proc. 14th Brazilian Symp. Computer Graphics and Image Processing*, pp. 179–182, 2001.

[3] A. M. Alattar, E. T. Lin, and M. U. Celik, "Digital watermarking of low bit-rate advanced simple profile MPEG-4 compressed video," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 13, no. 8, pp. 787–800, Aug. 2003.

[4] I.J. Cox, J. Kilian, F.T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673–1687, Dec. 1997.

[5] F. Deguillaume, G. Csurka, J. O'Ruanaidh, and T. Pun, "Robust 3d dft video watermarking," *SPIE: Security and Watermarking of Multimedia Contents*, Jan. 1999.

[6] E. T. Lin and E. J. Delp, "Temporal synchronization of video watermarking," *IEEE Trans. Signal Proc.*, vol. 52, no. 10, pp. 3007–3022, Oct. 2004.

[7] O. Harmanci and M. K. Mihcak, "Motion picture watermarking via quantization of pseudo-random linear statistics," in *Visual Communications and Image Processing 2005. Proceedings of the SPIE*, July 2005, vol. 5960, pp. 1142–1150.

[8] M. K. Mıhçak, R. Venkatesan, and M. Kesal, "Watermarking via optimization algorithms for quantizing randomized statistics of image regions," *Proceedings of the Fortieth Annual Allerton Conference on Communication, Control and Computing*, Oct. 2002.

[9] M. Kucukgoz, O. Harmanci, M. K. Mıhçak, and R. Venkatesan, "Robust video watermarking via optimization algorithm for quantization of pseudo-random semi-global statistics," *Proc. SPIE*, 2005.