Efficient Integration of Watermarking With MPEG Compression

Ju Wang
Computer Science Department
Virginia Commonwealth University
Email: jwang3@vcu.edu

Adrian R. Steele and Jonathan C.L.Liu
Computer, Info Science and Engineering Department
University of Florida
Email: jcliu@cise.ufl.edu

Abstract—This paper proposed a video watermarking scheme which embed watermarking directly in the compressed domain, thus can be deployed in real-time scenario. Our method embeds pseudo-random sequence to the qualified AC coefficients for a number of selected 8 by 8 DCT blocks. The effects of motion compensation introduced by the video compression is carefully considered to reduce the watermark-interference, a phenomena which can significantly affect the detection performance if not taken care of. Our scheme shows a low computation overhead and compatible robustness under moderate video compression.

I. INTRODUCTION

With the fast growing of high speed access to the INTERNET and multimedia streaming technologies, copyright protection of digital media has become a major concern for content provider to support large scale media-on-demand system. This had imposed an urgent demand for reliable digital watermarking technologies, especially for image and video data. Many researches had shown that watermark in spatial/frequency significant region [1]–[4], [7] is quite resilience to some general attack. However, there are only a few works in literature [2], [3], [8] discussing video watermark. It is realized that simply treating video as a sequence of still image and applying the traditional image watermarking is not efficient and might even expose the host media under some new type of attacks. This is largely due to availability of huge number of marked images. The redundant watermarking information in many similar video frames present a potential hole for malicious pirates, and are generally weak against accumulation attack.

In general, embedded watermarks can be designed as data and/or software/firmware code that intentionally embedded into the MPEG-2/4 content. However, no matter which methods are used, computing watermarks does take CPU time, thus generates resource contention with streaming delivery. Therefore, there are design tradeoffs between correlation detection (i.e., robustness), complexity (i.e., computing time), and other performance metrics. The issues become even more challenging when we need to support a large number of concurrent streaming accesses with different embedded watermarks. Since video titles are usually recorded and transported in compressed format (e.g., MPEG-2/4 format), the embedding needs to be able to work with compressed video with minimal computation overhead.

For compressed video, hiding watermark information in DCT coefficients [8], Run-Length-codes and motion vector have been discussed. However, detection for the last two method is very difficult in the decompressed domain. Hartung’s method [8] works for uncompressed and compressed video, but the watermark energy in their method seems distributed evenly in the block-transformed DCT domain, thus is not optimal for most video compression. Wu [2] and Podilchuk [7] proposed to use image local properties of human visual system in hiding data in the block transformed image/video. They did not address embedding in the compressed domain and the effect of motion compensation.

We believe the key design should be how to apply the various watermarking schemes effectively. For this purpose, we evaluated the performance for several video watermarking schemes with MPEG-2 compression. We found that the computing overhead of these methods is significant for broadcast quality video. We thus focused on a method based on 8 by 8 DCT blocks, which is more computation flexible by changing the number of marked blocks. However, if the number of marked blocks is too few, the degree of detection correlation will be reduce. Our preliminary results suggest that 100 marked blocks will be a minimum for robust detection.

An interesting observation during the investigation is that the P/B frame have a noticeable smaller detection response than I frame, especially when the bit budget is low (say less than 2 mbps). We found that due to the motion compensation in the MPEG-2 compression, the watermark signal of the predicting frame will be propagated to the downstream frame, causing a degradation of detection performance. This necessitates special design consideration when embedding watermark for P/B frame. To effectively mark the predicted blocks, we described a fast drift compensation where at most four macroblocks in the predicting frames needs to be calculated for each marked block. When the number of watermarked blocks decrease, the saved computation is considerable. The experiment results confirmed that the robustness of our scheme is comparable to the reference model. Compared to the conventional schemes, our proposed scheme intends to speed up the watermarking process with a sufficient number of selected blocks. In the following, we present the performance evaluation of Cox and JAWS schemes in section 2, and the proposed method in section 3.

II. PERFORMANCE CONSIDERATION

A. Cox’s Scheme Evaluation

One straightforward method of embedding watermark in video is to treat video frame as a sequence of still image and apply Cox’s method directly in the raw video frame.
domain. We implemented Cox’s method, and Fig 1(a) shown the robustness measurement of Cox’s method under MPEG-2 compression. A 30-frame test video is watermarked frame-by-frame with Cox’s method, and compressed under 5 mbps and 2 mbps. The detection is performed upon the reconstructed video frames. The video resolution is 704*480. We generally found that the detection response is quite good in normal situation. For 5 mbps case, except for the third frame, all other frames results in a 90% correlation to the watermark signature. The detection performance for 2-mbps case also show a strong correlation to the original watermark. An exception is in the third frame, where both case have a relative lower correlation value. Our interpretation is that, due to the imperfect rate control in MPEG-2 encoding, the bits allocated for the third frame is significant lower than other frames, which results in a high loss. This behavior is also observed for the other frames: I-frame always has a highest detection response, P-frame has the second high, B-frame usually have lowest detection response. Nevertheless, Cox’s method is robust enough to survive moderate low MPEG-2 compression (2 mbps for broadcasting level video).

Unfortunately, Cox’s method requires too much computation time to be implemented in a real-time scenario. We measured the executing time for inserting a 100-points watermark into different size of the video on a 2.4 GHz P4 LINUX machine. For a 512 by 512 resolution, Cox’s method use 0.30 seconds in the insertion. For a 2048*2048 video size, it takes 5.137 seconds. We also notice that the a long watermark sequence only introduce little more executing time. Inserting a 10000-points watermark results in a 0.15% increase in execution time, compared to the 100-points case. Thus the majority computation is spent in calculation the forward DCT coefficients.

Fig. 1: Detection after MPEG-2 coding (a)Cox’s method, compressed at 2 mbps and 5 mbps, (b) JAWS algorithm: compressed at 2 mbps

B. JAWS Evaluation

We also implemented and evaluate the performance of JAWS scheme [3] with MPEG-2 compression in our software simulation environment. For more details on JAWS scheme, readers are refereed to [3]. Basically, the JAWS scheme embeds watermark information to the 8 by 8 pixel blocks. Robustness is achieved via temporal redundancy and tilting the same 8 by 8 watermark matrix through the same picture. After embedding, the marked frames are encoded with same parameter as the previous test. For a fair comparison, coherent detection is performed here. Fig 1(b) shows the watermark detection for JAWS algorithm with MPEG-2 encoding. We observed that: (1) The peak detection response corresponds to the 1st and 16th frame, which are all intra-coded frame (I-frame). (2) The detection response for the the P-frames is in the mid-range, yet still distinguishable. (3) However, the detection response for B-frames is much worse than I/P frames, mostly below 0.15 after normalization. These are not sufficient as a positive detection evidence.

We believe that MPEG-2 encoding algorithm affects the survivability of JAWS watermark in two distinct ways:(1)the motion prediction, especially bi-directional prediction for B-frames, additively bring watermark information from other frame to local frame as a noise source, thus weaken the desired local watermark; (2) The current rate control scheme in MPEG-2 tends to over-allocate bits for the beginning few frames and under-allocate for the last few frames when the bit budget is not sufficient, which often results in a more severe watermark damage for the frames in the end of GOP. These two effects in general will weak all watermark scheme. The JAWS scheme seems to suffer more than Cox’s method, due to the fact that JAWS watermark is locally embedded in the pixel domain, which is more subject to quantization noise.

III. PROPOSED SCHEME

In our proposed scheme, the MPEG-2 stream is partially parsed to extract the DCT blocks as in [8]. Using the block selection algorithm discussed later, a predefined number of DCT blocks are identified for embedding. To watermark a 8 by 8 DCT block, we use an additive embedding method proposed by [7]. The resultant watermarked DCT blocks are written into the bitstream along with the MPEG-2 header/sider information. The computation in the above procedure contains the following major steps: MPEG-2 header parsing, run-length decoding for the DCT coefficients, de-quantization, adding the DCT transformed watermark signal, quantization, and run-length encoding. These steps are applicable for all frames. For P and B type frame, additional steps are necessary to mark non-intra-coded blocks.

A. Basic Embedding: I-frame

The embedding of watermark for the I-frame can be achieved similar to the IA-DCT scheme as suggested in [7]. Using their terminologies, this procedure can be expressed as following:

\[
X_{u,v,b}^* = \begin{cases} 
X_{u,v,b} + \frac{R_{u,v,b}}{X_{u,v,b}} w_{u,v,b} & \text{if } X_{u,v,b} > \frac{R_{u,v,b}}{X_{u,v,b}} \\
X_{u,v,b} & \text{otherwise}
\end{cases}
\]

Here \(X_{u,v,b}\) and \(X_{u,v,b}^*\) is the original and watermarked AC coefficients respectively, the watermark sequence \(w_{u,v,b}\) follow the N(0,1) Gaussian distribution. \(R_{u,v,b}\), also known as JND, is the embedding depth for different frequency components calculated according the human visual system [6].

Since the embedding depth is fully determined by the DCT coefficients of the 8 * 8 block, the above embedding procedure can be performed in the compressed domain with partially
decoded frame. For the sake of convenience, this embedding scheme is referred as IMODE in the rest of this paper.

The corresponding detection procedure consists of: (1) extract the difference between original and suspect frame in the (block) transformed DCT domain, \( w_{\alpha \nu} = X_{\alpha \nu} - X_{\alpha \nu} \), (2) the difference is then normalized, \( u_{\alpha \nu} = \frac{w_{\alpha \nu}}{\sum_{\alpha \nu}} \). Since the original frame is available in the detection party, the normalization factor, which is the embedding depth \( \gamma_{\alpha \nu} \), can be exactly calculated. (3) after this, the similarity based detection statistic \( p = \frac{u_{\alpha \nu} \cdot w_{\alpha \nu}}{\sum_{\alpha \nu}} \) is calculated.

The correlation detection response for intra-coded frames is evaluated for a test MPEG-2 stream coded at 5 mbps. By varying the number of watermarked DCT blocks, we are able to describe the relation between watermark length \( n \) and detection performance. It is observed that the detection response quickly become saturate when \( n \) is sufficiently large. When \( n \geq 50 \), the similarity between extracted watermark and the original stables at 0.75 or higher, which is a strong evidence of presence. Our experiments also confirmed that the detected correlation response is lower than 0.1 if the suspect video frame is not marked by the testing watermark sequence.

B. Design Issues in Inserting/Extracting Watermark for Predicted Frame

Watermarking for the predicted blocks is not so straight. One problem is so-called drift compensation problem described by Hartung in [8] and recently addressed in MPEG-4 coding by Alattar [5]. However in both works, watermark signature is embedded in the pixel domain and the choice of embedding depth (or amplitude factor \( \alpha \)) did not fully utilize the HVS. The problem of drift compensation becomes more severe when watermark is embedded according to HVS. In order to correctly embed watermark signature for these blocks and be extracted later, we need the complete block information during embedding, which should be obtained by combining the prediction block and residue information. Otherwise, if only residue or prediction information is used as the basis of embedding, the normalization factor (embedding depth) cannot be extracted.

To show this, assume that we apply IMODE to the residue block without considering the motion compensation. The embedding depth at embedding time is thus calculated from the residue DCT block. Unfortunately, at the detection party we usually have raw video images, and the estimated embedding depth is based on the complete block information, which is usually significantly different to those used in watermark embedding time. As a matter of fact, the JND value is proportional to the block average luminance [6]. Realizing that the residue block is often significantly dark than the original block, the JND value based on the residue block will be much smaller than the JND value based on the original frame block. Using JND based on original frame to normalize the extracted watermark can only results in much weak correlation response, thus a higher probability of detection errors. Our experiments indicates that the JND calculated from the frame residue information is magnitude smaller than its true value, and the contribution of the watermark effectively become zero after re-quantization.

Emulating MPEG-2 encoding and generate approximated residue information can also be erroneous, since the implementation of MPEG-2 encoder is not standardized and different codec might results in varying motion prediction decision and generate different residue block information. The button-line is, at the watermark embedding, the embedding depth for predicted blocks in P/B frame should be calculated from the corresponding original blocks. A straight method to do so, is to perform a full scale MPEG-2 decoding for the P and B frame, transform the reconstructed picture back to DCT domain, insert watermark with IMODE, decompress the modified DCT blocks back to pixel domain, and re-encode the watermarked picture into P or B frame. This brute-forth approach, however, requires much more computation and are not suitable for the video-on-demand scenario.

C. Embedding Procedure for Predicted Blocks in Predicted Frames

For the sake of convenient in discussion, we define some useful notations in table I.

![Fig. 2](attachment:image.png)

**Table I:** Some notations used for motion drift compensation

The spatial relationship between the predicted DCT blocks \( I_{\alpha \nu} \) and its reference block \( I_{\alpha \nu} \)'s are illustrated by figure 2. Without lose of generality, we only discuss the case for a forward coded block here. Thus \( I_{\alpha \nu} \) is in the current P-frame, and \( I_{\alpha \nu} \)'s can be regarded as blocks in the reference I-frame. The discussion in this subsection can be easily extended to the bi-directional predicted blocks in B-frame with minor modifications.

![Fig. 2](attachment:image.png)
Let $\Phi_1(k,h)$ be the watermarked version of $I_1(k,h)$. Based on the discussion of previous subsection, to compute $I_1(k,h)$, we need the residue block $R_1$ and the prediction block $I_0$ from the predicting frame. Specifically, we have

$$I_1(k,h) = I_0(x,y) + R_1(k,h)$$  \hspace{1cm} (2)

$R_1$ is readily obtained after run-length decoding for the current frame, however, $I_0(x,y)$ is not. It is frequently observed that $(x,y)$, the location of the best-matched-block in the predicting frame for $I_1(k,h)$, is not aligned to the 8 by 8 boundary as shown in figure 2. Since all coded blocks are aligned to the 8 by 8 grid, the non-aligned predicting block $I_0(x,y)$ can not be extracted from the compressed bitstream directly. To calculate $I_0(x,y)$, the four surrounding blocks $I_{0}^{U}, I_{0}^{L}, I_{0}^{R}, I_{0}^{D}$ and $I_{0}$ should be decoded to provide the surrounding area containing $I_{0}(x,y)$, then a 8 by 8 luminance block can be extracted to calculate the forward DCT $I_{0}(x,y)$.

Having $I_1(k,h)$ available from (2), the local JND matrix $J_1(k,h)$ can also be calculated according to [6]. The modulated watermark is then $W_1(k,h) = J_1(k,h)(z)\Phi_1(k,h)(z)$. It should be pointed that the actual predicting block, denoted by $\Phi_0(k,h)$, used at decoder is the ‘watermarked’ version of $I_0(x,y)$. We emphasis the word watermark to indicate the fact that $\Phi_0(k,h)$ is a join-effect of the watermarks on the surrounding blocks. Here $\Phi_0(k,h)$ can be calculated similar to $I_0(x,y)$, except that all operation is based on the watermarked predicting frame.

Finally, we have the modulation algorithm for the residue block in P frame

$$\hat{R}_1(k,h) = \Phi_1(k,h) - \Phi_0(x,y)$$  \hspace{1cm} (3)

$$= I_0(x,y) + R_1(k,h) + W_k,h - \Phi_0(x,y)$$

$$= R_1(k,h) + W_k,h + (I_0(x,y) - \Phi_0(x,y))$$

D. Block Selection and Detection Synchronization

Our preliminary experiments show that the detection performance converge to a very high level as the number of marked block become sufficiently large (more than 50). Thus the overall computation overhead can be controlled by only marking a small portion of the DCT blocks, say the best first 100 DCT blocks, instead of all 5000 some blocks. In order to do this, we should be able to identify the same set of best blocks at both embedding and detection time. We found that the block variance is a good measurement to choose suitable blocks. A large variance usually indicates a complex pixel pattern, large AC coefficients in mid-range, and thus can bear more watermark noise. Furthermore, block selection can be performed off-line and the results be saved as block mask for future use.

Since the block mask is highly image dependent, it is possible to use the block mask to assist image and synchronization. Let $M_i$ be the block mask for the $i$th frame and $M^*$ be the block mask for the suspect video frame, a simple method to decide the original frame is to calculate the correlation measurement among $M$’s and $M^*$ and choose the one with highest correlation. This method has shown satisfactory positioning capability in our experiments. We has used two GOP from different scenes as the source video. These video frames are MPEG-2 encoded at 5 mbps bitrate, watermarked in the compressed domain, and decoded back to. The detection simulator then calculate the block mask for the reconstructed video frames (we did not need the source video in this calculation), which is correlated to the block masks of the source video. The results shows that the correlation is generally small (less then 0.2) when the suspect video frame is not in the same video scene as the candidate original video. When the two video frames in test is from the same video scene (e.g., the same GOP), we always observed the highest correlation value (0.55 to 0.7 in this experiment) at the perfect alignment point. The correlation value with other frames in the same GOP is between 0.24 to 0.47.

The initial results from our simulation is encouraging. In Fig.2(b), the detection response for 30 frames from one testing video is presented. The video is encoded at 5 mbps and 3 mbps respectively, and 50 DCT blocks from each frame are marked. All cases result in a correlation value acceptable as a positive evidence (we set a threshold of 0.1 ). Thus our proposed scheme can provide reasonable watermarking for moderate low MPEG-2 compressed video. The quality degradation of video is not visible compared to the non-watermarking case. At 5 mbps encoding bitrate, when the number of watermarked block is less than 200, the PSNR for the decoded image maintain higher than 37 db for all video titles in test. The lowest PSNR is about 32 db when all blocks are marked.

IV. CONCLUSION

We described a novel video watermarking scheme for realtime embedding compressed video. Our scheme embeds watermark sequence to the 8 * 8 DCT blocks based on human visual system to limit the visual disturbance. Drift compensation is performed with partially decoded MPEG-2 stream, and a block selection mechanism is used to facilitate the realtime embedding. The initial simulation results is encouraging.

REFERENCES