Content-Based Block Watermarking Against Cumulative and Temporal Attack

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

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

***Abstract*— This paper presents a block-selection-based video watermarking scheme that is designed to be resilient against two dangerous attacks: cumulative attack and temporal attack. We use content-based block selection to counteract cumulative attack by spreading the locations of marked blocks. The block selection algorithm also leads to a novel frame synchronization method that can effectively re-synchronize suspected video frames to their original positions. Our scheme has low computation overhead and robust detection performance for moderately compressed video.**

I. INTRODUCTION

Image watermarking [1] [4] and video watermarking [2], [3], [5] are becoming increasingly important in copyright protection. For block-compressed video, hiding watermark information in DCT coefficients [5], run-length-codes, and motion vectors has been discussed. However, detection for the last two methods is very difficult in the decompressed domain. In [5], motion drift compensation technique is used to combat the watermark cross-talk caused by the motion-prediction during compression/decompression.

However, most of the existing schemes do not take into consideration the unique property of video watermarking: the strong correlation between successive video frames. Due to large numbers of marked images, the redundant watermarking information in many similar video frames might be exploited by malicious pirate attack. In [6], collusion attack is analyzed, where collections of video frames are combined to generate an unmarked copy of the original.

We discuss a similar collusion attack that is related to drift-compensated block-based watermarking, referred to as *cumulative attack* in the context of this paper. Such an attack can be easily launched if video frames are watermarked by the same signature at the same block locations [1]. Furthermore, if watermarked blocks are poorly selected, the cumulative attack could readily reveal the watermark signature even though the block location is not known to the attacker. The selection of the watermarked block thus should be carefully designed to be immune to cumulative attack. Our proposed method adopts a per-frame block selection algorithm to dynamically select image blocks for watermark embedding. Each frame is analyzed based on its local image content and its dependency to the I-frame. With this content-based block selection approach, the effects of watermark interference are further reduced.

Based on the proposed block selection algorithm, we develop a frame synchronization method that is robust against temporal attack, where the attackers purposely destroy the original frame sequence to compromise the watermark detection. The dependency of watermark location on the local image properties provides ancillary information for frame synchronization at the detection phase. The synchronization algorithm is optimized to reduce the detection time by employing a two-stage synchronization: a rough search to locate the right Group of Picture (GOP), and a refined search to lock the suspect frame down to the exact position in the original video. Experimental results confirm that the robustness of our scheme is comparable to the reference model and the two types of aforementioned attacks are successfully blocked.

The rest of this paper is organized as follows. Section 3 shows how image blocks should be selected based on the image content in order to deal with the cumulative attack. Section 4 presents the frame synchronization method that effectively combats the temporal attack and the experiment results. Section 5 concludes our work.

II. BLOCK WATERMARKING AND MOTION COMPENSATION

This section provides a review of the block watermarking and motion compensation technique. The basic structure of our proposed scheme follows that of [5]: the MPEG-2 stream is partially parsed, coefficients of the DCT blocks are modified as necessary and written back into the bitstream. During this process, we introduce a novel block selection algorithm to trade off the computation complexity, video fidelity, and watermark robustness. To minimize the impact to the image quality, watermarks for an individual 8 by 8 DCT block uses an additive embedding method [4].

We only show the embedding procedure for a forward predicted block. Denote by $I_{1,i}$ the i^{th} 8 by 8 block at the current frame, $R_{1,i}$ the 8 by 8 residue DCT block corresponding to $I_{1,i}$, and $I_{0,i}$ the 8 by 8 predicting block for $I_{1,i}$ in the reference frame. Let $\Phi_{1,i}$ be the watermarked version of $I_{1,i}$, the z^{th} matrix element (after zig-zag scan) is

$$\Phi_{1,i}(z) = \begin{cases} I_{1,i}(z) + J_{1,i}(z)w_i(z) & \text{if } I_{1,i}(z) > J_i(z) \\ I_{1,i}(z), & \text{otherwise} \end{cases} \quad (1)$$

[1] In a practical detection system, using a unique watermark for different video frames is at least computationally deficient and requires a very powerful synchronization method

0-7803-9332-5/05/$20.00 ©2005 IEEE

where w_i is the i^{th} 8 by 8 watermark matrix and $J_{1,i}$ is the local embedding depth (JND) matrix calculated from $I_{1,i}$. To compute $I_{1,i}$, we need the residue block R_1 and the prediction block I_0 from the predicting frame. Specifically, we have

$$I_{1,i} = I_{0,i} + R_{1,i} \tag{2}$$

R_1 is readily obtained after run-length decoding for the current frame. $I_{0,i}$ is obtained from the decompressed I-frame. The desired residue block after watermarking and motion compensation is then

$$\hat{R}_{1,i} = \Phi_{1,i} - \Phi_{0,i} = R_{1,i} + J_{1,i}w_i + (I_{0,i} - \Phi_{0,i}) \tag{3}$$

III. Block Selection

Our preliminary experiments show that the detection response converges to a nearly 90% level as the number of marked blocks becomes sufficiently large (> 50). Thus, the overall computation overhead can be controlled by only marking a small portion of the DCT blocks, say 100 DCT blocks, instead of all 5000-plus some blocks. The question to be answered is: *how should one select image blocks for watermarking?* The trivial answer, to hide watermark information in the same predefined positions for all video frames, is very vulnerable to cumulative attack as shown next.

A. Threat from Cumulative Attack

The fixed-block-selection method results in a simple extractor design: always check the pre-defined blocks for watermark information. However, the simple design has a serious security drawback. Let us examine a scenario where M slow-motion video frames with static background are watermarked. These video frames are MPEG-encoded into one GOP.

Further assume that the watermark embedder selected a background block i in which to embed watermark bits. With static block selection, the same block is also selected for all P-frames. After motion prediction, the coded residue DCT matrix of block j at the j^{th} P frame is

$$\hat{\mathbf{R}}_j^i = \mathbf{R}_j^i + \Phi$$

Here $\mathbf{R}_j^i$ is the residue information of the original video frame, and $\hat{R}_j^i$ is the residue information after watermark insertion. Since block i is static background, it can be very well predicted and the energy of the block differences (residue blocks) is usually low. In fact, the true residue matrix $\mathbf{R_j^i}$ can be treated as an independent zero mean Gaussian matrix.

Thus, if the block selected for watermarking is known, a potential attacker can add these residue blocks to get

$$\hat{\Phi} = \frac{1}{M}\sum_j \hat{R}_j^i$$

It is easy to verify that $E[\hat{\Phi}] = \Phi$ and $Var[\hat{\Phi}] = \frac{1}{M}\sigma^2$ where σ^2 is the variance of the original background residue. Figure 1 shows the estimated watermark matrix via the above simple cumulative attack. The average $\frac{estimation}{actual}$ ratio is plotted for different σ^2 values as a function of video frames used in the attack. With a few video frames encoded this way, the watermark can be isolated easily. It should be pointed out that the above attack can be applied to any block that is prediction-coded. The better the motion prediction is done, the smaller the residue block energy will be, and the more likely those blocks will reveal the watermark signature. Such blocks are called *dangerous blocks*.

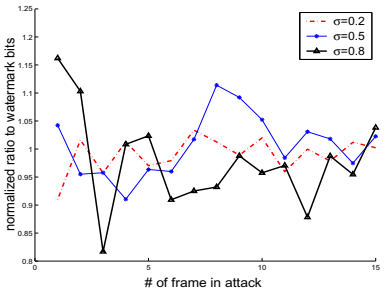

Fig. 1: Cumulative attack on predicted block when same block is used in a group of picture for watermarking

A smart attacker can even launch a cumulative attack without knowing the positions of watermarked blocks. The attacker can even utilize the encoded motion vectors to facilitate the search of dangerous blocks. For example, to check whether a block j is watermarked or not, the following algorithm can be used:

Potential Cumulative Attack: for K^{th} GOP

1) Find 6 consecutive P-frames where block i is prediction-coded and the residue energy is smaller than a threshold value, say $2\sigma^2$.
2) If such a frame sequence is found, calculate the average block and save it as B_k.

Calculate the correlation of all B_ks. If there is a strong correlation, it indicates a high probability that this block is watermarked. The average of the B_ks should be a close estimation of the embedded watermark signature. With a block analysis tool like the above algorithm, an attacker has a good chance to identify all watermark locations and a close estimation of watermark signature. Therefore static block selection is susceptible to attack.

B. Content Based Block Selection

To combat cumulative attack, a safe watermark scheme must distribute watermark signatures into different blocks along the video sequence. One solution is to select image blocks based on the image properties and hard-code the block selection algorithm in the extractor. [4] suggested selecting image blocks with rich contents. This was originally proposed to minimize the watermark-caused video quality degradation. We believe the content-based method might generate a dynamic block pattern which is desired to defeat cumulative attack. A good indication of image complexity is the block variance. However, a scheme solely based on local variance tends to result in the selection of many overlapping image blocks for consecutive video frames, especially when there is only little content change in video scene. Furthermore, if a video frame

contains complex background color, these background blocks will dominate the selected image blocks. These background image blocks are often well predictable from the I-frame and the resultant residue blocks have small intensity and variance, and are thus more vulnerable under a cumulative attack.

We suggest join consideration of the motion information and the residue block variance in the block selecting algorithm. The principle is to select image blocks that provide the greatest difference between the current video frame and its predicting frame. In this fashion, high priority is given to image blocks that can not be predicted from the previous video frame. These image blocks are usually newly-appearing video objects. If the video is relatively static and there is not a sufficient amount of intra-coded blocks with high variance, priority should go to image blocks belonging to moving front objects. Such image blocks often have a high motion vector. The selection algorithm is described by the following pseudo-code:

Motion-Variance Block Selection (MVBS)

1) Calculate the residue image by subtracting the predicting frame from the candidate video frame by using motion parameters.
2) For each block i in the residue image:
   a) calculate the variance value v_i,
   b) calculate the moving index $u_i = (1/8)\sum_{k\in neighbor}|m_k|$, here m_k is the motion vector of the block k.
   c) calculate the weighted quantity $U_i = w * v_i + (1 - w) * u_i$ where $0 < w < 1$ is a weight parameter specified by the individual algorithm. A video clip with slow motion should adopt a high w value.
3) Sort the image blocks in descending order of U_i. The first N blocks will be selected for watermarking.

Figure 2 shows the image blocks selected according to the above algorithm. The performance of the MVBS selection behavior is summarized below: (1) The MVBS algorithm successfully avoids the selection of simple smooth background, such as the blue sky background in the test video clips. This preserves the subjective video quality since the watermark noise is less *visible* if they are embedded in complex video objects. The risk of revealing watermark position through visual analysis is also reduced.

(2) Image blocks containing object boundaries are often selected. This is because such blocks contain pixels from different video objects that are quite different in color. In Figure 2, many selected blocks contain part of the contour of the house object and the lamp pole object. This trend becomes more apparent when we increase the number of blocks (see Figure 2.(b)).

(3) Figure 2.(c,d) show the selected block masks for two consecutive P-frames within the same GOP. Although the two frames are very similar to each other, the MVBS captures the subtle difference and generated very different watermark locations.

IV. Frame Synchronization

An important issue in watermark detection is that the watermarked video must be synchronized to the original copy. This

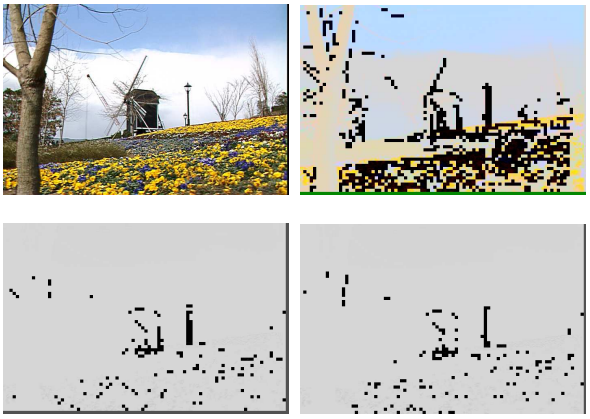

Fig. 2: Variance based block selection: (a) original video frame, (b) select the top 1000 blocks, (c) the top 100 blocks for the first P-frame, (d) the top 100 blocks for the second P-frame

is most critical when only a segment of video frames needs to be verified. Most video watermarking techniques assume that frame-level synchronization is established before detection. If this synchronization is destroyed, either by spatial distortion or temporal offset, the detector can no longer correctly detect the watermark.

Watermark synchronization against spatial distortion (such as geometrical transformation) has been discussed by many authors [8] [7]. However little has been mentioned regarding temporal synchronization. In fact, temporal domain attack is much easier to carry-out: an attack can simply drop a few frames from the watermarked video, and the damage in detection could be very severe. In Figure 3.(a), the average watermark detection performance of an off-synchronized video segment is shown. In this experiment, the same watermark is embedded into all video frames. We purposely use a video clip with slow motion. For each video frame, marked DCT blocks are individually selected with the MVBS block selection algorithm. At the detector side, we introduced an artificial frame offset to mimic the temporal attack. An offset d means the detector will use frame $i + d$ as the original for suspect frame i. The correlation response is low and the detector fails to declare a match even with a small offset d=1.

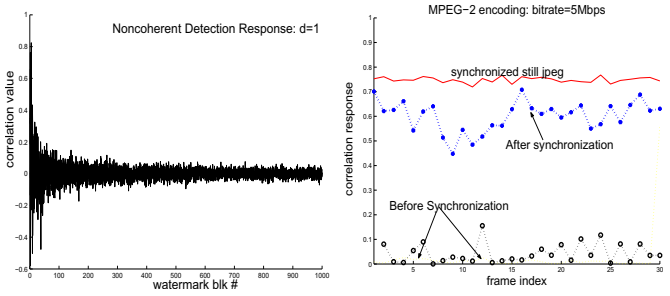

Fig. 3: (a) Detection response with temporal offset, (b) detection response after synchronization

The problem is that we are pursuing two seemingly conflicting goals: on one hand the watermark should be distributed in different locations among different video frames to deal with cumulative attack; on the other hand, the more dispersed that the watermark locations are, the more vulnerable it is against temporal attack.

Fortunately, the MVBS block selection method also suggests a way to synchronize the watermarked video frames back to the original video title. The idea is to re-construct the watermark locations of the suspect video frame, and synchronize this frame to an original frame having *the most similar* watermarked block location.

The watermark block locations for original frame i are now denoted as block mask $M(i)$, which can be calculated by the detector from the original copy. Let M' be the block mask of the suspect video frame. An estimation of $M^{'}$ is obtained by the following procedure:

Group of Picture Position:

1) Locate the video scene that contains the suspect video. This could be done most likely with human assistance. A completely automatic process might consume huge computation time (e.g., use frame-by-frame correlation check) and the accuracy is still not guaranteed.
2) Let $S_I = \{I_1, I_2, ...I_N\}$ be a set containing all N I-frames of the identified video scene. For each I-frame in S_I, the detector performs a motion prediction to encode the suspect video frame. The resultant residue frames form the set $S_R = \{R_1, R_2, ...R_N\}$.
3) Find the parameter j that minimizes the residue-frame-energy by:
$$j = \arg\min_i \|R_i\|$$
4) Use frame I_j as the predicting I-frame, and execute the MVBS algorithm for the suspect frame. The resultant watermark locations are used as an estimation of M'.

The above GPP algorithm will narrow the synchronization range down to the GOP level. After this initial search, there is a high probability that the suspect video frame is within the GOP headed by the identified I-frame I_j. Let $G(I_j)$ represent the original video frames in this GOP. The next step is to further identify the one frame within the given GOP such that the probability of exact match is high. This is accomplished by calculating the correlation measurements among $M(i)$'s (of $G(I_j)$) and M', the estimated block mask of the suspect frame. The last step is to choose the frame with the highest correlation. The estimated original frame f_{syn} to synchronize is then
$$f_{syn} = \arg\max_{k \in G(I_j)} corr(M(k), M')$$

This method shows satisfactory positioning performance in our experiments. We use several GOP from different scenes as source video to which suspect video frames will be synchronized. These video frames are MPEG-2 encoded at 5 mbps, and watermarked by the MVBS algorithm. The watermarked video frames are then decompressed and shuffled randomly to destroy the original order. For each of the re-shuffled frames, the detection simulator execute the synchronization algorithm and estimates a block mask, which is correlated to the block masks of the source video. We test the synchronization performance with 30, 50, and 100 marked blocks/frame. The average block mask correlation $c()$ is measured under each of the following three synchronization conditions: exact match (M), match with frames in the same GOP (MG), and match with other GOPs (NoM). With 30 blocks, the $c()$ values for the three conditions are 0.40, 0.37, and 0.13, respectively. Increasing the block number to 50 and further to 100 shows an steady increase of $c()$ in the exact match condition. Meanwhile the $c()$ values for MG and NoM conditions decrease monotonically. With 100 blocks, we observe 0.67, 0.25 and 0.06 for the three matching cases, showing an apparent separation among these cases. The GOP hit ratio, which is the probability that the initial search finds the right GOP for a suspect frame, is 100% in all cases and independent of the marked block number. With 100 marked blocks, our algorithm produces a correlation response margin of 0.30, which is sufficient to distinguish the synchronized point from other frames.

In Fig.3.(b), the watermark detection response for 30 randomly selected watermarked frames is presented. The video is encoded at 5 mbps, and 100 DCT blocks from each frame are marked. The watermark correlation is calculated after frame synchronization. All cases result in a correlation level that is acceptable, and therefore validates our methodology (the detection threshold is set to 0.20). Thus, our proposed scheme can provide robust watermark detection performance under cumulative attack and temporal attack.

V. CONCLUSION

We propose a content-based block selection algorithm to counteract the cumulative attack by varing the marked block location. The block selection algorithm also leads to a frame synchronization method that can effectively re-synchronize the suspect frame to its original position.

REFERENCES

[1] Ingemar J. Cox, Joe Kilian, F. Thomson Leighton, and Talal Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Trans. on Image Processing*, vol. 6, no. 12, Dec 1997, pp. 1673–1687.

[2] Min Wu, Heather Yu, and Bede Liu, "Data Hiding in Image and Video: Part II–Designs and Applications," *IEEE Trans. on Image Processing*, vol. 12, no. 6, June 2003, pp. 696-705.

[3] T. Kalker,"A Video Watermarking System for Broadcast Monitoring," *Proceedings of SPIE, Security and Watermarking of Multimedia Contents*, vol.3657, San Jose January 1999, pp.103-112.

[4] Christine I. Podilchuk, and Wenjun Zeng, "Image-Adaptive Watermarking Using Visual Models," *IEEE Journal on Selected Areas in Commnications*, vol.16, no.4, 1998, pp. 525-537.

[5] F. Harhung and Bernd Girod, " Watermarking of Uncompressed and Compressed Video," *Signal Processing* vol.66, no.3, 1998, pp.283-301

[6] Karen Su, Deepa Kundur, and Dimitrios Hatzinakos. "A content-dependent spatially localized video watermark for resistance to collusion and interpolation attacks," In Proc. International Conference of Image Processing, 2001. pp. 818-821.

[7] Shih-Wei Sun and Pao-Chi Chang, "Video watermarking synchronization based on profile statistics", *IEEE Aerospace and Electronic Systems Magazine*, vol.19, no.5, 2004, pp. 21-25.

[8] P-C Su, et.al., "Synchronized Detection of the Block-based Watermark with Invisible Grid Embedding", Proceedings of SPIE, Security and Watermarking of Multimedia Contents III, vol.4314, January 2001, pp. 406-417.