Real-Time Compressed Video Watermarking in the VLC Domain

Chun-Shien Lu (呂俊賢)*, Jan-Ru Chen (陳建儒)**, Hong-Yuan Mark Liao (廖弘源)*, and Kuo-Chin Fan (范國清)**

*Institute of Information Science, Academia Sinica, Taipei, Taiwan, ROC
**Dept. of Computer Science and Information Engineering, National Central University, Chung-Li, Taiwan, ROC
e-mail: {lcs, philips, liao}@iis.sinica.edu.tw, kcfan@csie.ncu.edu.tw

Abstract

This paper proposes a compressed domain video watermarking scheme for copyright protection. In addition to common requirements of watermarking, the unique features of video watermarking such as preservation of bit-rate, compressed domain embedding/detection, and video attacks are examined. For real-time watermark detection, our method is directly applied in the MPEG2 bitstream. More specifically, watermarks are inserted into the variable length codeword (VLC) domain. We shall discuss how to select proper data in a video bitstream to embed watermarks while preserving perceptual fidelity. The power of our method is reflected by its robustness capability against attacks. False positive probability is analyzed to confirm our assertion.

Keywords: Watermarking, Protection, Robustness, Blind detection, MPEG, VLC, Bit-rate.

1. Introduction

Digital watermarking has been proposed to be a very useful technology in the protection of digital data such as image, audio, video, formatted documents (PDF or PS), and 3D objects. In the literature [3], most of the existing watermarking approaches are conducted on images. However, video is even more useful and should be protected with higher priority. In particular, video sequences usually contain rich properties that images do not have. On the other hand, the types of attacks applied on a video are much different from those applied on an image. In this paper, we will focus ourselves specifically on video watermarking.

In the past, a number of video watermarking schemes have been proposed [1, 2, 4, 5, 7, 8, 9, 10]. The existing video watermarking schemes either conceal watermarks in a raw video [1, 2, 7, 8, 9, 10] or in a compressed video [2, 4, 5]. Compressed video watermarking methods are considered more practical because video sequences always exist in a compressed form. For compressed video watermarking, Hartung and Girod [2] proposed a spread spectrum-based oblivious video watermarking scheme. They arranged a watermark sequence to be a two dimensional array with the same size as a video frame. Then, the watermark signal is $8 \times 8$ DCT transformed and added into the DCT coefficients of a video stream. Therefore, some preprocessing operations such as inverse entropy coding and inverse quantization are required before watermark embedding. Their detection process was proven to be very close to real time. Although the bit error rate (BER) of [2] was analyzed, it is really surprising that the performance of robustness against different attacks was not evaluated.

In [4], Langelaar et al. proposed a video watermarking scheme in the compressed domain based on variable length codewords (VLC). At first, they
divided run-level pairs into a finite number of groups with the same VLC codeword length under the constraint that the level difference in each group should be exactly $1$. During watermark embedding, a run-level pair is either unchanged or replaced by the other codeword in the same group. Using this arrangement, the video's bit-rate will not be increased after embedding. Besides, the difference of level between the incoming codeword and the selected codeword is fixed to be $1$. This implies the modification of video's quality is almost negligible. Although Langelaar et al.’s work [4] is simple and efficient, the robustness of their scheme is not high.

In [5], Langelaar et al. proposed a differential energy watermarking (DEW) algorithm performed in the DCT domain. In their approach, an I-frame (or an image) is divided into blocks and each block is also divided into two sub-regions. The energies of the two sub-regions are, respectively, calculated by a cut-off point index that indicates the positions of zigzag-scanned DCT coefficients. This constant cut-off point index indicates a trade-off control between robustness and visibility. Langelaar et al. chose a larger cut-off index for the judgment of video transparency. For watermark embedding, a watermark bit is inserted by setting one of the energies of sub-regions to zero. That is, part of the high-frequency DCT coefficients will be eliminated. The authors claimed that it is not possible to remove the DEW watermark without causing perceptual degradation.

From the above review, we know that a robust real-time video watermarking scheme is a must. However, almost all compressed video watermarking methods were, in fact, performed in the `DCT’ domain. In other words, both inverse entropy coding and inverse quantization must be performed in advance before watermark embedding or detection. In this work, we shall propose a compressed-domain video watermarking method, which can be directly performed in the VLC domain while satisfying the requirements of transparency, robustness, and blind detection. Only inverse entropy coding needs to be performed in our watermarking method. In the proposed method, we take two major issues into consideration. First, the concept of viewing watermarking as communication with side information will be employed [6], as described in Sec. 2. Second, how to design a video watermarking scheme entirely operated in the compressed (VLC) domain is described in Sec. 3. We will also consider how to embed watermarks into a video sequence without increasing its bitstream size because increase of bit-rate is not permitted in practical situations. In Sec. 4, experimental results together with false positive analysis have demonstrated the superiority of our video watermarking system.

2. Watermarking as Communications with Side Information

In [6], we employed the communications with side information concept into our image watermarking scheme and found that good robustness and acceptable fidelity could be achieved. We derived that hidden watermarks could be reliably extracted in a blind manner if mean filtering was considered in the design level. Basically, our watermarking scheme is accomplished by replacing the difference between the original data and the mean filtered (the predicted original) data with the hidden watermark. Please refer to [6] for more details.

For compressed video watermarking, we propose to embed one watermark into each video frame (regarded as an image). Owing to a video sequence is usually compressed by local DCT transform, we will modify our previous image watermarking scheme [6] to embed watermarks in the block-based transformed domain. As to how to apply this modified technique and make it
appropriate for video watermarking is elaborated in the following.

3. Compressed Video Watermarking

In this section, we shall describe how to select suitable data at suitable positions for video watermarking. We will also discuss how to not increase the bit-rate of a video sequence after watermark embedding. More importantly, false positive analysis is provided to measure the performance of our method.

3.1 Selection of Suitable Data

In the video decoding stage, the video bitstream is usually decoded into variable length codewords (VLC). Every codeword corresponds to a run-level pair denoted as \((r, l)\). For a given run-level pair \((r, l)\), the value of \(r\) indicates the positions of a set of DCT coefficients in a zigzag-scan order. In addition, the value of \(l\) represents the magnitude of a DCT coefficient only. More specifically, run \(r\) represents the number of DCT coefficients with magnitude zero preceding the current run-level pair and level \(l\) corresponds to the quantization value of the current DCT coefficient. If a run value \(r\) is changed for current codeword, then the frequency in a zigzag-scan order is also significantly changed. The frequencies of succeeding codeword are also significantly changed. In other words, the image would be distorted significantly because a great number of DCT coefficients have been changed. On the other hand, if a level value \(l\) is changed, then only one DCT coefficient will be changed. It will not affect other frequency. Therefore, we understand that it is much easier to preserve the fidelity of a video by simply modifying the level values \(l\). Based on the above reasons, we have chosen to embed watermarks by modulating level values \(l\) instead of run values \(r\).

3.2 Selection of Suitable Position

In this section, we consider what kind of GOP structure in a compressed bitstream is suitable for watermarking. Basically, a GOP structure is composed of three kinds of frames: “I”, “P”, and “B.” Owing to the data amount of the “B” and “P” frames are relatively fewer than that of the “I” frame, we decide to embed watermarks in an “I” frame only. Embedding watermarks in an I-frame has the advantage of avoiding quality degradations. In addition, the watermark hidden in an “I” frame could be propagated into other “B” and “P” frames so that watermarks can still be detected from those non-“I” frames.

Since the size of a video bitstream is subject to change at different compression ratios, the number of run-level pairs will also be changed. This will cause the asynchronization problem when embedding and detecting watermark values at different positions. In order to tackle this problem, we propose to conceal a watermark value into a macroblock (MB) because the number of macroblocks is invariant against different attacks including compressions. In addition, the watermarks are inserted into the Y component of I-frames so that the problem due to sampling can be avoided. In this paper, the length of a hidden watermark is equal to the number of macroblocks. The diagram of the proposed MB-based embedding process is shown in Fig. 1.

![Fig. 1: The proposed MB-based watermark embedding process.](image)
3.3 Video Watermark Embedding

Suppose there are in total $N$ macroblocks in a video bitstream. In the $i$-th macroblock, let $(r_{ij}, l_{ij})$ be the $j$-th run-level pair, $u(i)$ be the mean of levels, and $n_i$ be the number of levels. Under these circumstances, the mean values $u(i)(1 \leq i \leq N)$ will form a 1-D sequence. Let $\bar{u}(i)$ be the mean filtered value obtained from $u(i)$. We can obtain the difference values: $d(i) = u(i) - \bar{u}(i)$. Next, these $d(i)$ values can be Gaussian normalized into $d_G(i)$ with zero mean and unit variance. Then, a watermark sequence $w = \{w(1), w(2), ..., w(N)\}$ to be hidden is generated as a Gaussian distribution with zero mean and unit variance. Every watermark value $w(i)(1 \leq i \leq N)$ is embedded into a macroblock by replacing its corresponding $d_G(i)$ value such that the modified $d_H(i)$ is equal to $w(i)$ [6]. Therefore, the modified $d(i)$ value is obtained as

$$d_H(i) = w(i) \cdot \sigma_d + u_d = d_G(i) \cdot \sigma_d + u_d,$$  \hspace{1cm} (1)

where $u_d$ and $\sigma_d$ are the mean and standard deviation of $d(i)$.

So far, watermark embedding is actually not finished because only the quantity used to modulate the mean of level values, $d_H(i) - d(i)$, is obtained in a macroblock. We still need to modulate every run-level pair in a macroblock. In this paper, we propose to propagate the modulation quantity $d_H(i) - d(i)$ to all levels of the run-level pairs in a macroblock. That is, the original run-level pair $(r_{ij}, l_{ij})$ is modulated as $(r_{ij}, l_{ij}^H)$ with

$$l_{ij}^H = l_{ij} + (d_H(i) - d(i)),$$  \hspace{1cm} (2)

where $1 \leq j \leq n_i$ and $1 \leq i \leq N$.

Owing to the mean filtering is used to extract the signal from a cover data [6], it is not difficult to know this propagation will still satisfy Eq. (1). This implies that we have changed from macroblock-based modulation to level-based modulation.

However, such kind of modification (Eqs. (1) and (2)) still needs to be further improved because the replacement of $d_G(i)$ by $w(i)$ might cause visual defects, i.e., significant modifications of DCT coefficients will occur. In order to maintain fidelity, a transparency constraint should be imposed. In the VLC domain, the maximum quantity that could be changed on a ‘level’ value should be constrained to be exactly 1. Therefore, the following proposed transparency constraint should be satisfied:

$$|d_H(i) - d(i)| = 1 = |w(i) - d_G(i)| \cdot \sigma + \mu |.$$  \hspace{1cm} (3)

Based on this constraint, we can further derive the practically embedded watermark value, $w^e(i)$, as

$$w^e(i) = \begin{cases} d_G(i) + \frac{1 - \mu_d}{\sigma_d}, & \text{if } w(i) \geq 0, \\ d_G(i) - \frac{1 + \mu_d}{\sigma_d}, & \text{if } w(i) < 0, \end{cases}$$

in order to preserve $w(i)$ and $w^e(i)$ to have the same sign. Under this paradigm, the modified $d_G(i)$ value should be

$$d_H^e(i) = w^e(i) - d_G(i).$$  \hspace{1cm} (4)

Besides, the modified $d(i)$ value is changed from the form in Eq. (1) to be

$$d_H^e(i) = \begin{cases} d(i) + 1, & \text{if } w(i) \geq 0, \\ d(i) - 1, & \text{if } w(i) < 0. \end{cases}$$  \hspace{1cm} (5)

Accordingly, the $u(i)$ value will be modulated as

$$u^H(i) = \bar{u}(i) + d_H^e(i).$$  \hspace{1cm} (6)

Using Eq. (5) instead of Eq. (1) will automatically consider the trade-off between transparency and robustness.

During the process of compressed video watermarking, we have found some problems that should be particularly addressed. If these problems are not suitably solved, the requirements of transparency and robustness will not be achievable. If $(r_{ij}, l_{ij}^H)$ does not exist in the VLC codewords, then it will create the
video coding and decoding problem. In order to avoid this problem, the modulated \((r_{ij}, l_{ij})\) should be enforced to stay with the value of its closest run-level pair in the VLC codewords. Besides, if the difference between the original level value and its corresponding modulated level value is large enough, the fidelity of the video would be hard to be preserved. Therefore, we have chosen to maintain the original run-level pair unchanged. On the other hand, if the modulated level \(l_{ij}^h\) is less than or equal to zero, then it is set to be 1 in order to maintain the correctness of decoding.

After video watermarking, we should deal with the bit-rate (BR) problem. In a compressed video sequence, the size of a bitstream is not allowed to increase after watermarking, but decrease in bitstream’s size is permitted. It is very easy to insert dummy bits into a watermarked bitstream to maintain the size unchanged if a bitstream has been decreased in size. Preservation of bit-rate is particularly important for video watermarking in the compressed domain. Therefore, a trick is proposed to achieve the above goal. We first check the total amount of size reduction by embedding negative watermark bits. Then, the total amount of size increased by embedding positive watermark bits must be controlled to be smaller than the amount of size reduction.

### 3.4 Video Watermark Detection

From a suspect video, we first calculate its mean level values and mean filtered level values of macroblocks as \(\bar{u}(i)\) and \(\bar{u}'(i)\) for \(1 \leq i \leq N\), respectively. Then, the extracted watermark values could be determined as

\[
w^e(i) = u^a(i) - \bar{u}'(i),
\]

which is basically an inverse operation of Eq. (5).

Finally, a normalized correlation value derived between the original watermark \(w(i)\) and the extracted watermark \(w^e(i)\) is defined as

\[
\rho = \frac{\sum_{j=1}^{N} \text{sign}(w(j)) \cdot \text{sign}(w^e(j))}{N},
\]

where the sign function in Eq. (8) is defined as

\[
\text{sign}(t) = \begin{cases} 
1, & \text{if } t \geq 0, \\
-1, & \text{if } t < 0.
\end{cases}
\]

This correlation value is used to indicate the existence of a hidden watermark.

### 3.5 False Positive Analysis

It is critical to compare the detection value with a pre-determined threshold in order to judge the existence of a watermark. Usually, false positive analysis of a video watermarking system should be conducted, in particular, in the case of DVD player. False positive detection means a watermark is detected but in fact there is no watermark hidden. Imagine that one wants to make a copy of his/her own home video and the watermark detector mistakenly thinks that content is copy protected and no copying is allowed. Under these circumstances, one as a consumer would be very inconvenient and upset. We would rather be permissive first (allow one to make a copy which one really shouldn't) then be restrictive later (forbid one to copy which one really should be able to do).

Based on the concept of Central Limit Theorem, in this paper we assume that the detection values obtained from either watermarked videos or non-watermarked videos form a Gaussian distribution. Therefore, it is required to determine a threshold \(T\), which can separate these two distributions with a negligible false positive probability. Let \(\mu_\rho\) and \(\sigma_\rho\) be the mean and standard deviation of the detection values \(\rho\) obtained from non-watermarked videos. Let \(\varepsilon\) be a desired false positive probability. Therefore, the false positive probability could be obtained as

\[
\varepsilon = P(\rho < T) = Q\left(\frac{T - \mu_\rho}{\sigma_\rho}\right),
\]

where

\[
P(\rho < T) = \int_{-\infty}^{T} f_\rho(x) \, dx
\]

and

\[
Q(x) = \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} e^{-\frac{1}{2}t^2} \, dt.
\]
where $Q(\cdot)$ is the $Q$-function and is defined as a tail integral of a unit-Gaussian probability density function, and $\rho_{rv}$ is a random variable having distribution of $\rho$. Suppose the desired false positive probability is $\varepsilon = 10^{-6} (10^{-7})$, one can easily derive the threshold as $T = 0.137 (0.150)$ by using our experimental data: $\mu_\rho = -0.004$ and $\sigma_\rho = 0.0288$.

4. EXPERIMENTAL RESULTS

Our watermarking scheme was tested on several famous video sequences, which were compressed at 15 Mbit/s. In addition, the MPEG-2 codec [11] was adopted for video encoding and decoding. The results obtained by testing two video sequences, ‘flower-garden’ and ‘table-tennis,’ will be shown here. The size of a frame in the experiments was $704 \times 576$. The PSNR values of the watermarked video frames were higher than 36 dB, as shown in Figs. 2 and 3. Perceptually, the original video and the watermarked video were visually indistinguishable. This implies that our watermarking scheme is able to achieve video transparency.

Some commonly adopted attacks were used to test the robustness of our method. These attacks included MPEG compression with bit-rates of 6M, 4M and 2M bps, additive noise adding ($= 27$ dB), sharpening, frame averaging, frame rate changing, and I-frame dropping+compression. In addition, the correlation values obtained by extracting watermarks from some non-watermarked videos were also calculated. In Figs. 4 and 5, we show the correlation values detected from the attacked ‘flower-garden’ and ‘table-tennis’ videos, respectively. The horizontal axis indicated the frame number and the vertical axis indicated the correlation values. From Figs. 4, 5 and the derived threshold $T$ in Sec. 3.5, we can see that the correlation values detected from those watermarked/attacked videos were easily separated from those detected from un-watermarked videos. This implies that both the false negative and the false positive probabilities obtained from our method are low.

In order to test the robustness of our scheme, we conducted another experiment by assuming attackers know in which (“I” frames) we embedded our watermarks. Fig. 6 shows the results detected from the ‘flower-garden’ sequence with all original “I” frames dropped. It can be seen that some detection values went down significantly. These phenomena occurred when large motions emerged and hence the hidden
watermarks cannot be sufficiently propagated.

To demonstrate that our watermark detection can be accomplished in real-time, the time required for three different scenarios were compared and shown in Fig. 7. The three scenarios included (1) the process of decoding and re-encoding, (2) watermark detection in the compressed video domain (our work) plus decoding, and (3) decoding directly. The first scenario indicated the situation when the watermark should be detected in the spatial domain. Therefore, a compressed video must be decoded for detection and then re-encoded to recover its original data. The second scenario indicated that our method detected watermarks in the compressed domain directly and the detection time was added back together with the decoding time. The third scenario was used to measure the baseline time for traditional video decoding. This comparison was conducted on the Pentium-III 700 machine. In Fig. 7, it can be observed that the execution time of (2) and (3) were very close to each other. This implies that the execution time of our watermark detection process is negligible and can be said very close to being real-time.

5. CONCLUDING REMARKS

In this work, a new compressed video watermarking scheme has been presented based on the concept of communications with side information. We have proven that our scheme could satisfy the requirement of real-time detection.

From the current results, we have found that the correlation value detected from a watermarked (but not
attacked) video was not very high. This is because around 20% ~ 30% of the macroblocks either have all zero level values or generate new codewords that do not exist in the VLC table. Note that level values are in fact corresponding to DCT coefficients if inverse quantization is applied. If the embedding is conducted on some small DCT coefficients, then the watermarked video won't be robust. Therefore, our ongoing work focuses on improving the robustness by embedding watermarks on some selected run-level pairs so that watermark bits can be really inserted.

![Time complexity comparisons](image)

**Fig. 7.** Time complexity comparisons: (1) decoding and re-encoding; (2) decoding and compressed video detection (our work); and (3) decoding directly.

**REFERENCES**


