Real-Time Frame-Dependent Watermarking in MPEG-2 Video *

Chun-Shien Lu*, Jan-Ru Chen†, and Kuo-Chin Fan†

* Institute of Information Science, Academia Sinica, Taipei, Taiwan, ROC

† Department of Computer Science and Information Engineering,

National Central University, Chung-Li, Taiwan, ROC

Abstract

Digital watermarking is a helpful technology for providing copyright protection for valuable multimedia data. In particular, video watermarking deals with several issues that are unique to various types of media watermarking. In this paper, these issues, including compressed domain watermarking, real-time detection, bit-rate control, and resistance to watermark estimation attacks, will be addressed. Since video sequences are usually compressed before they are transmitted over networks, we first describe how watermark signals can be embedded into compressed video while keeping the desired bit-rate nearly unchanged. In the embedding process, our algorithm is designed to operate directly in the variable length codeword (VLC) domain to satisfy the requirement of real-time detection. We describe how suitable positions in the VLC domain can be selected for embedding transparent watermarks. Second, in addition to typical attacks, the peculiar attacks that video sequences encounter are investigated. In particular, in order to deal with both collusion and copy attacks that are fatal to video watermarking, the video frame-dependent watermark (VFDW) is presented. Extensive experimental results verify the excellent performance of the proposed compressed video watermarking system in addressing the aforementioned issues.

Keywords: Attack, Bit-rate control, Collusion, Video frame-dependent watermark, Real-time detection, Robustness

Corresponding author: Chun-Shien Lu (email: lcs@iis.sinica.edu.tw)

*This paper is a significantly modified and extended version of an earlier paper [16] that was published in the Proc. of IEEE Int. Conf. on Communications, France, 2004.

1 Introduction

1.1 Literature Review

Due to the rapid development of the Internet during the past decade, numerous methods have been proposed for storing and transmitting digital multimedia data. Digital data is much superior to analog data because the quality of copies is not degraded at all. However, this superiority has become a threat to authorized usage because digital data can be easily tampered with, duplicated or distributed without the need for expensive tools. As a consequence, the intellectual property protection problem has become an urgent issue in the digital world.

Recently, an emerging intellectual property protection scheme called "digital watermarking" has been extensively explored [4, 7, 19]. Digital watermarking is a helpful technique that can be used to protect multimedia data. The underlying concept of digital watermarking is to embed imperceptible signals into digital multimedia data to carry out specific missions. When watermarked multimedia data encounters a digital signal processing operation, the embedded watermark is expected to survive this operation. If someone violates the copyright of watermarked multimedia data, the original owner can prove ownership by extracting the embedded watermark. In this paper, we shall focus on the development of a new video watermarking scheme.

In the past decade, a number of video watermarking schemes have been proposed. The existing video watermarking schemes embed watermarks either in raw video [3, 5, 6, 9, 22] or in compressed video [2, 6, 11, 12, 13]. Compressed domain video watermarking is considered more practical and is the objective of this paper because video is usually stored in a compressed format before it is transmitted over networks. In [6], Hartung and Girod proposed a video watermarking scheme that uses MPEG-2 bitstreams as the underlying target data. They arrange a watermark sequence into a 2D format, which is the same size as a video frame. Then, the watermark signals are 8×8 DCT transformed and added into their corresponding DCT coefficients. In order to deal with the error propagation problem caused by watermarking, they add a drift compensation signal to the watermarked signal. Their detection process has been shown to be very close to real-time detection. Arena *et al.* [2] improved Hartung and Girod's work with interleaved encoding. They also make use of the advantageous properties of the human visual system (HVS). In [11], Langelaar *et al.* proposed a video watermarking scheme which can be used in the compressed domain by modifying the codewords generated by a variable length codeword (VLC). First, they divide run-level pairs into many groups, where each group contains codewords of equal length and the level difference in each group is exactly one. During watermark embedding, a run-level pair

is either left unchanged or replaced, depending on the incoming watermark value. Their method is basically a least significant bit (LSB) approach. This implies that quality degradation of a video after embedding can be almost negligible. However, for a commonly adopted signal processing operation, such as decoding followed by re-encoding, achieving robustness becomes impossible. In [12], Langelaar *et al.* proposed a differential energy watermarking (DEW) algorithm which can be employed in the DCT domain. They calculate the energy difference of two blocks and remove the high frequency coefficients of the corresponding block with energy being pre-defined as smaller in order to maintain the pre-defined energy relation. The authors mentioned that their method is robust against the re-encoding of video bit streams. However, as mentioned in [1, 12], this technique is susceptible to transcoding, especially when the GOP (group of picture) structure is changed. An earlier paper [22] described a scene-based video watermarking approach that uses a "temporal wavelet transform." One major feature is that the authors first pointed out the collusion attack, which is a common threat to video watermarking methods. In fact, the collusion attack can be said to be an unique outcome [†] of video watermark methods. We deeply believe that it would be meaningless to claim that a video watermarking scheme was robust if it could not deal with collusion. Overall, although the aforementioned papers have dealt with problems related to video watermarking, some issues still remain. In order to give the reader a clear idea of the state of the art in video watermarking, Table 1 summarizes selected techniques versus video characteristics. In the following subsections, the issues of compressed domain watermarking, real-time detection, bit-rate control, and resistance to peculiar attacks that video sequences may encounter will be discussed.

1.2 Compressed Domain Video Watermarking, Real-Time Detection, and Bit-Rate Control

As for watermarking in the compressed domain, most of the existing compressed domain video watermarking methods are, in fact, employed in the "DCT" domain. In other words, both inverse entropy coding and inverse quantization must be performed before watermark embedding or detection. In this work, we shall propose a compressed-domain video watermarking method in which the watermarking process can be directly performed in the VLC domain. In comparison with the existing DCT domain watermarking methods, the places where we propose to embed watermarks are closer to the coded bit-stream. The advantage of this embedding strategy is its efficiency in real-time detection. With the

[†]It is worth mentioning that if each image unit (e.g., a block or mesh) in an image is treated like a frame in a video sequence; then collusion attacks can also be applied to those image watermarking methods that employ a multiple redundant watermark embedding strategy [15].

watermark embedded in the VLC domain, tedious DCT and inverse DCT transforms can be avoided. On the other hand, since the codeword length can be calculated by looking up the VLC table, the bit-rate control problem can be easily handled.

1.3 Robustness of Video Watermarking with Emphasis on Resistance to Collusion and Copy Attacks

As for the problem of resistance to video attacks, it is known that robustness is the critical issue affecting the practicability of any watermarking method employed in a DRM system. The robustness of the current watermarking methods has been frequently examined with respect to removal attacks or geometrical attacks or both. Removal attacks try to eliminate the hidden signal $\mathbf{W}$ (originally embedded in the cover data $\mathbf{I}$) by manipulating the stego data $\mathbf{I^s}$ such that the fidelity of the attacked data $\mathbf{I^a}$ is inevitably destroyed (i.e., $PSNR(\mathbf{I}, \mathbf{I^s}) \geq PSNR(\mathbf{I}, \mathbf{I^a})$). However, there also exist attacks that can defeat a watermarking system without sacrificing perceptual quality. Typically, the collusion attack [15, 21, 22], which is one of the removal attacks, can make colluded media data more perceptually similar to its cover version (i.e., $PSNR(\mathbf{I}, \mathbf{I^s}) \leq PSNR(\mathbf{I}, \mathbf{I^a})$). The collusion attack, in particular, is fatal to video watermarking. Collusion attacks in video watermarking are divided into two types [22]: Type I collusion attacks (applied to video frames embedded with the same watermark) and Type II collusion attacks (applied to video frames embedded with different watermarks). A Type I collusion attack is conducted first by averaging a set of extracted watermarks (usually obtained through using denoising) to estimate the hidden watermark, and then the estimated watermark is subtracted from all the frames in order that the hidden signal can be removed, whereas a Type II collusion attack is operated by averaging those perceptually similar frames in order to directly remove the watermarks. However, Type II collusion is less powerful since it is restricted to operating only on a subset of video frames such that the video watermarks cannot be eliminated entirely. Hence, we will only focus on the Type I collusion attack in this paper. It should be noted that the conventional denoising-based removal attack [23] when applied to one single image is a special case of the collusion attack.

On the other hand, the copy attack [10], which is one of the protocol attacks, is used to create the false positive problem. Initially, the copy attack was proposed in image watermarking and carried out as follows: (i) a watermark is first predicted from a stego image; (ii) the predicted watermark is added into a target image to create a counterfeit stego image; and (iii) from the counterfeit image, a watermark can be detected that wrongly claims rightful ownership. Compared with the collusion attack, the copy attack can be executed on only one video frame or an image and, thus, is more flexible. In this regard, the copy

attack must not be ignored when robustness is mentioned. Because watermark estimation is a common step performed to realize the collusion and copy attacks, they will be called watermark-estimation attacks (WEAs).

In the literature, previous collusion-resistant video watermarking methods were either computationally complex [22] or dependent on unstable feature extraction [21]. In addition, the copy attack has been ignored. In this paper, we propose a content-dependent video watermarking scheme to resist both collusion and copy attacks. First, each video-frame hash is extracted and then combined with a hidden message to yield the video frame-dependent watermark (VFDW). The properties of the VFDW will be examined, and mathematical analyses of the VFDW's resistance to WEAs will be discussed.

The remainder of this paper is organized as follows. In Sec. 2, we shall study how to embed and blindly extract watermarks in the VLC domain, and how to keep the bit-rate of the resultant stego video nearly unchanged. In Sec. 3, the design and analyses of the proposed video frame-dependent watermark in resisting both collusion and copy attacks will be discussed. Extensive experimental results will be given in Sec. 4, and concluding remarks will be drawn in Section 5.

2 MPEG-2 Bitstream Watermarking

The proposed compressed domain MPEG-2 video watermarking scheme will be discussed in this section. We shall start with a discussion of how to embed watermark signals in the VLC domain. Then, a macroblock-based embedding scheme will be described. We will also investigate the problem of bit-rate control. Finally, we will describe how the hidden watermark can be blindly extracted. In this paper, "cover video" is used to denote an MPEG-2 compressed bitstream, and "stego video" is used to denote a (compressed+embedded) bitstream.

2.1 Video Watermarking in the VLC Domain

When an MPEG bitstream is to be watermarked, it is required to be decoded backwards to some extent. In our method, only inverse entropy coding must be executed since our system is applied in the VLC domain. After a cover (compressed) video is inversely entropy coded, an MPEG compressed bitstream is represented using variable length codewords, as tabulated in Table 2. In the VLC table, each codeword corresponds to a run-level pair, denoted as (r, l). During a video encoding stage, the pixel values in the spatial domain are 8x8 Discrete Cosine Transformed (DCT). After quantization, the content of each 8x8 block is scanned in a zigzag manner. Each scanned non-zero integer has to be converted into a

so-called run-level pair, (r, l). The run, r, indicates the number of zeros preceding the current non-zero coefficient. The level, l, on the other hand, represents an integer that is the magnitude of the non-zero coefficient after quantization. Having the run-level pairs for all the non-zero quantized coefficients in an 8x8 block, one can consult the VLC table and convert them into a bitstream. It is easy to see that if one chooses run-level pairs as potential hiding places, then it will be feasible to modify the level value only. Based on the MPEG coding rule, if a run value is modulated, then this indicates that the number of preceding zeros has been altered. Basically, this kind of update will lead to serious consequences because the positions of all subsequent non-zero coefficients will be shifted. Under these circumstances, if these shifted coefficients are decoded back to the spatial domain, the resultant frame will be quite different from its original. However, if we update the level value, then only its magnitude will be changed. Under these circumstances, visual degradation due to watermark embedding can be controlled. Therefore, we propose to update the level value instead of updating the run value during the watermark embedding process.

There is a potential problem if one adopts level values as hiding places. It is known that if the compression ratio is changed, then the size of a video bitstream will be changed as well. Under this circumstance, the number of run-level pairs will also be changed. The above change will result in the so-called asynchronization problem, which will lead to misdetection of the hidden watermark because the position of the watermark has been shifted. In order to tackle this problem, we propose to use a macroblock (MB) as an embedding unit because the number of macroblocks within a frame can be kept constant if the frame size is not changed after attacks are applied.

Another issue regarding video watermarking is the selection of a proper color channel for embedding. Usually, the frames of a video sequence will be split into Y, Cb and Cr channels in the MPEG coding stage. According to different color resampling rules, it is possible for the ratio Y:Cb:Cr be set to 4:2:2 or 4:2:0. Under these circumstances, the only unchanged channel is the Y channel. Thus, when embedding watermarks, we prefer to exploit the Y channel as the host channel. In addition to the above selections, choosing an appropriate frame type among I-frame, P-frame or B-frame for hiding watermarks is also a crucial issue. Usually, a conventional video consists of a number of GOPs (group of pictures). Each GOP is composed of one I-frame and several B-frames and P-frames. A typical I-frame adopts intra coding, which means it does not refer to any other frames. Different from an I-frame, a P-frame only refers to its nearest preceding I- or P-frame. As for a B-frame, it refers to the nearest preceding and succeeding I- or P-frame. In a conventional MPEG format, the content of a B- or P-frame is the so-called residual error between the current frame and the frame to which it refers. Therefore, only an I-frame can hold

complete information. In this paper, we choose to embed watermarks into the I-frames of an MPEG compressed video sequence. Due to the inherent referencing effect, we know that watermarks embedded in I-frames will be propagated into succeeding B- or P-frames when re-encoding is performed. In this situation, watermarks can be detected from all the frames. In the next section, we shall present the detailed procedures for embedding and extracting watermarks.

2.2 Macroblock-based video watermarking

In addition to different (spatial/transformed/compressed) domain watermarking methods, in order to deal with geometrical distortions, the embedding of synchronization patterns/templates in advance for later recovery of geometrical parameters has been popularly used in image watermarking and offers a certain benefits. In the literature, regularly tiled subframes [8] or feature point-based irregular subframes [21] have been proposed as a basic embedding unit to provide a certain degree of resistance to geometrical attacks. In [1], Alattar *et al.* also implemented synchronization patterns and embedded them into a MPEG-4 bitstream to solve rotation and scaling problems. Of course, we can also adopt similar principles to handle geometrical attacks in our system. However, as we have pointed out in [15] and will describe later in Sec. 3, embedded synchronization/repetition patterns can be easily removed by means of the collusion attack. As a result, we would rather focus on those attacks that are peculiar to video sequences.

In this work, we shall propose a content-dependent, collusion- and copy-resilient blind video watermarking system, where the original source is not used during the detection process. The rationale behind using blind detection is mainly based on the fact that the intrinsic content of a video can be mostly preserved even when digital operations are applied. Here, the so-called "intrinsic content" of a video is defined as its filtered version since the noisy part is discarded by means of mean filtering. It is known that a watermark signal is usually a high-frequency signal; therefore , watermark embedding and detection steps are operated in the noisy part. In the following, we shall describe in detail the macroblock-based video watermarking scheme. A block diagram of our method is illustrated in Fig. 1.

2.2.1 Video Watermark Embedding

Suppose there are, in total, N macroblocks in a video frame. Let (r_{ij}, l_{ij}) be the j-th run-level pair in the i-th macroblock, let $u(i)$ be the mean of the levels in the i-th macroblock, and let n_i be the number of levels in the i-th macroblock. Under these circumstances, the mean values $u(i)$ $(1 \leq i \leq N)$ will form a 1-D sequence $\mathbf{U}$ for each video frame. Let $\bar{u}(i)$'s be the mean filtered version obtained from $u(i)$'s. Since $u(i)$ is the mean level value of a macroblock, its corresponding mean filtered value $\bar{u}(i)$ in the mean

filtered sequence $\bar{\mathbf{U}}$ can be derived by averaging the mean level values of its left and right neighbors; i.e.,

$$\bar{u}(i) = \frac{1}{mf_s} \sum_{k=i-\lfloor \frac{mf_s}{2} \rfloor - 1}^{k=i+\lfloor \frac{mf_s}{2} \rfloor} u(k) \tag{1}$$

is obtained and mf_s denotes mean filtering support. By applying this mechanism to all the macroblocks, we can obtain a set of magnitude relationships between each pair of $u(i)$ and $\bar{u}(i)$ as

$$mr(i) = sgn(u(i) - \bar{u}(i)), \tag{2}$$

where $sgn(\cdot)$ is a sign function and is defined as

$$sgn(t) \quad = \quad \begin{cases} +1, & \text{if } t \geq 0, \\ -1, & \text{if } t < 0. \end{cases}$$

Through this procedure, the noisy part, the $(u(i) - \bar{u}(i))$'s, of a host signal, from which the watermark will be embedded, is filtered. In addition, we assume that the approximate version, $\bar{\mathbf{U}}$, of a host signal can be mostly obtained in the watermark detection process in order to not affect robustness under blind detection.

Next, a watermark signal $\mathbf{W} = \{w(1), w(2), ..., w(N)\}$ is generated based on a secret key K for the purpose of embedding, where $w(i) = +C$ or $-C$, and C denotes a constant. In this paper, watermark embedding is done by using a watermark value $w(i)$ $(1 \leq i \leq N)$ to perturb $u(i)$ as

$$u^h(i) = u(i) + w(i), \tag{3}$$

where $u^h(i)$ is the modulated version of $u(i)$. After applying Eq. (3) to all the block mean values $u(i)$'s, each $\bar{u^h}(i)$ can be derived according to Eq. (1) as

$$\bar{u^h}(i) = \frac{1}{mf_s} \sum_{k=i-\lfloor \frac{mf_s}{2} \rfloor - 1}^{k=i+\lfloor \frac{mf_s}{2} \rfloor} u^h(k) = \frac{1}{mf_s} \sum_{k=i-\lfloor \frac{mf_s}{2} \rfloor - 1}^{k=i+\lfloor \frac{mf_s}{2} \rfloor} u(k) + \frac{1}{mf_s} \sum_{k=i-\lfloor \frac{mf_s}{2} \rfloor - 1}^{k=i+\lfloor \frac{mf_s}{2} \rfloor} w(k) \approx \bar{u}(i), \tag{4}$$

based on the assumption that $\frac{1}{mf_s} \sum_{k=i-\lfloor \frac{mf_s}{2} \rfloor - 1}^{k=i+\lfloor \frac{mf_s}{2} \rfloor} w(k) \approx 0$ due to the fact that the watermark values are constant and mf_s is an even integer. Therefore, the new magnitude relationship following embedding between $u^h(i)$ and $\bar{u^h}(i)$ is

$$mr^h(i) = sgn(u^h(i) - \bar{u^h}(i)). \tag{5}$$

Sequentially substituting Eqs. (3), (4), and (2) into Eq. (5), we get

$$mr^h(i) = sgn(u^h(i) - \bar{u^h}(i)) = sgn((u(i) + w(i)) - \bar{u}(i))) = sgn((u(i) - \bar{u}(i)) + w(i)). \tag{6}$$

Comparing Eqs. (2) and (6), it becomes clear that our embedding idea is to change the magnitude relationship, $mr(i)$, between $u(i)$ and $\bar{u}(i)$ according to the incoming watermark bit $w(i)$. In this paper, the goal of our embedding process is to force the modulated magnitude relationship (i.e., $mr^h(i)$) to have the "sign" the same as its corresponding watermark bit, $w(i)$. With this embedding rule, we have the following: (i) when $w(i)$ is positive, $u(i)$ must be modulated by adding $w(i)$ in order to get positive $mr^h(i)$; (ii) when $w(i)$ is negative, $u(i)$ must be modulated by adding $w(i)$ in order to get negative $mr^h(i)$.

However, the entire embedding process at this stage has not been carried out completely because watermark embedding only proceeds to the "macroblock" level. In practice, each run-level pair in a macroblock still needs to be modulated. Furthermore, we have not yet discussed the fidelity problem that is encountered following watermark embedding. With regard to level-wise modulation, we propose to propagate the modulation quantity $w(i)$ to all the levels of the run-level pairs in a macroblock i. This implies that all the original run-level pairs in a macroblock i are modulated with the same quantity; i.e., the level value l_{ij} is modulated as i_{ij}^h by means of

$$l_{ij}^h = l_{ij} + w(i), \tag{7}$$

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

As for the fidelity issue, since the embedding step (Eq. (7)) may produce visual defects, the actual watermark value $w(i)$ needs to be further studied here. Usually, a human visual model is adopted to maintain fidelity in digital watermarking. For VLC domain video watermarking considered here, the maximum quantity change allowed for a "*level*" value is "1," which corresponds to a fixed quantization interval in the DCT domain. As a consequence, the following transparency constraint needs to be enforced during the embedding process:

$$|l_{ij}^h - l_{ij}| = |w(i)| = 1. \tag{8}$$

Based on Eq. (8), the hidden watermark $\mathbf{W}$ is derived as a bipolar signal; i.e., the constant C is set to be 1. The magnitude of the watermark values exactly indicates the maximum distortion that we can impose on the level values of a video bitstream's variable length codewords.

During the compressed domain video watermarking process, however, we encounter some problems that should be carefully handled. For example, if the modulated run-level pair, (r_{ij}, l_{ij}^h), does not exist in the VLC codewords, then this will cause a video encoding and decoding problem. In order to tackle this problem, the modulated (r_{ij}, l_{ij}^h) should be forced to possess the level value of its closest run-level pair in the VLC codewords. This also implies that the transparency constraint, specified in Eq. (8), may not be satisfied when the aforementioned problem occurs. On the other hand, if the modulated level l_{ij}^h is

less than or equal to zero, then we propose to leave the original run-level pair unchanged to maintain the correctness of re-encoding. Under this circumstance, robustness will be more or less affected. In practice, our extensive results have shown that the watermark detection results obtained from stego/attacked video sequences can be separated very well from those obtained from un-watermarked video sequences.

2.3 Bit-Rate Control

Since adding watermarks into a video bitstream will destroy the structure of video data redundancy and thereby reduce coding efficiency so that the bit-rate of a stego video will be undesirably increased, in some applications, the bit-rate of a video bitstream can not be (dramatically) changed after a watermarking process is performed. Here, we shall explain how the problem of bit-rate control can be dealt with. According to the VLC table (Table 2), the number of bits used to represent a run level pair satisfies the following inequality:

$$VLC_{length}(r, l) \leq VLC_{length}(r, l + 1), \tag{9}$$

where the function $VLC_{length}(\cdot, \cdot)$ reports the number of bits used to represent the VLC codeword of a run level pair. Eq. (9) implies that given the same run r, a larger level value is associated with a longer codeword. Based on the above coding rule, we shall describe in the following how the bit-rate can be kept unchanged. Strictly speaking, our goal is to make the bit-rate difference between a cover and a stego video sequence negligibly small.

During the watermark embedding process, we first modify the level values of those macroblocks that correspond to negative watermark bits. Since negative watermark bits are to be embedded, the codeword length of a modulated level value will be shorter than that of an original level value, as indicated in Eq. (9). We will calculate the number of bits, $\mathcal{B}$, that have been saved after the negative watermark bits are embedded. Then, the increase in the bit-rate that results from embedding positive watermark bits must be kept smaller than or equal to $\mathcal{B}$. To this end, we propose to embed the positive watermark bits one by one until the increased bit-rate satisfies the above constraint. If there still are positive watermark bits that have not been embedded, we give up on embedding them in order to not increase the bit-rate. Thus, robustness will be affected. However, our extensive results have shown that the watermark detection results obtained from the stego/attacked video sequences can be separated very well from those obtained from un-watermarked video sequences. Through this embedding strategy, the bit-rate (BR_{stego}) of a stego video can be guaranteed to be equal to or smaller than that (BR_{cover}) of its corresponding cover video, i.e., $BR_{stego} \leq BR_{cover}$. In fact, the difference between BR_{stego} and BR_{cover} is negligibly small, as indicated by our experimental results.

2.3.1 Video Watermark Extraction

The watermark signal extraction process is fast and simple, and is basically an inverse process of embedding. Because watermark extraction is conducted in the VLC domain, inverse quantization, inverse DCT, and decoding+re-encoding are not performed on the incoming video bitstreams. On the contrary, only inverse entropy coding is required to obtain VLC codewords. Numerical results will be given later to show that real-time detection can be achieved.

In the watermark extraction process, the first step calculates the block-based mean level sequence, $\mathbf{U^s}$, and its mean filtered version, $\bar{\mathbf{U}}^{\mathbf{s}}$, as described in Sec. 2.2.1, of a suspect video. Since video attacks may have been imposed on a stego video, the resultant sequence, $\bar{\mathbf{U}}^{\mathbf{s}}$, will be not the same as its corresponding original, $\bar{\mathbf{U}}$. However, as explained previously, what we can rely on is the invariance of the intrinsic content (i.e., $\bar{\mathbf{U}}$) of a video. Strictly speaking, this assumption (i.e., $\bar{\mathbf{U}} = \bar{\mathbf{U}}^{\mathbf{s}}$) cannot be guaranteed and is only correct to some extent. However, our results indicate that this assumption is reasonable. After the two sequences $\mathbf{U^s}$ and $\bar{\mathbf{U}}^{\mathbf{s}}$ are obtained, each element $w^e(i)$ of an extracted watermark sequence $\mathbf{W^e}$ is determined as follows:

$$w^e(i) = sgn(u^s(i) - \bar{u}^s(i)) \approx sgn(u^s(i) - \bar{u}(i)) = w(i), \tag{10}$$

where $\bar{u}^s(i) \approx \bar{u}(i)$ is simply assumed to realize blind watermark estimation. In practice, $u^s(i)$ can be further formulated as the result of imposing a fading effect plus a noise component on $u(i)$. Then, the subsequent task is to solve the parameters of fading and noising, as previously discussed in [24] and other works.

In order to determine the presence/absence of a hidden watermark, the normalized correlation value between $\mathbf{W}$ and $\mathbf{W^e}$ is computed as follows:

$$\delta_{nc} = \frac{\mathbf{W} \cdot \mathbf{W^e}}{\sqrt{|\mathbf{W}||\mathbf{W^e}|}} = \frac{\mathbf{W} \cdot \mathbf{W^e}}{N}, \tag{11}$$

where "$\cdot$" is an inner product operation. The relationship between $\delta_{nc}(\cdot,\cdot)$ and BER (bit error rate) can be expressed as $\delta_{nc}(\cdot,\cdot) = 1 - 2 \times BER$. It is said that a watermark has been detected if $\delta_{nc}(\cdot,\cdot)$ is larger than a pre-determined threshold T. In this study, T was selected as 0.11 if the desired false positive probability was approximately 10^{-7} [4].

3 Video Frame-Dependent Watermark

In this section, we will describe the proposed video frame-dependent watermark (VFDW) that is embedded in our system. In Sec. 3.1, we shall first discuss the characteristics of watermark estimation

attacks (WEAs). Then, the proposed video frame hash and video frame-dependent watermark will be described in Sec. 3.2. Finally, the properties of the VFDW will be studied and its resistance WEAs will be analyzed in Secs. 3.3 and 3.4, respectively. Note that in order to better explain the resistance of VFDW to WEAs, our analyses will be conducted in the spatial domain. This is reasonable because a signal embedded in the transformed domain can be transferred to another equivalent signal in the spatial domain and watermark estimation by means of denoising [10, 23] is intuitively applied in the spatial domain. As shown in Table 1, the resistance of the video frame-dependent watermark to WEAs is the novel contribution of this paper.

3.1 Watermark Estimation Attack

From an attacker's perspective, the energy of each watermark bit must be accurately predicted so that the previously added watermark energy can be completely subtracted to accomplish effective watermark removal. An estimated watermark's energy is closely related to the accuracy of the removal attack. Several scenarios are shown in Fig. 2, which illustrates the energy variations of (a) an original watermark; (b)/(d) an estimated watermark (illustrated in gray-scale); and (c)/(e) a residual watermark generated by subtracting the estimated watermark from the original watermark. From Fig. 2(a)$\sim$(c), we can see that even though watermark's sign bits are fully obtained, the corresponding energies cannot be completely discarded, and the residual watermark still suffices to reveal the encoded message. Furthermore, if the sign of an estimated watermark bit is different from its original sign (i.e., $sgn(W(i)) \neq sgn(W^e(i))$), then any additional energy subtraction will not be helpful in improving removal efficiency. On the contrary, watermark removal in terms of energy subtraction operated in the opposite (wrong) polarity will undesirably damage the media data's fidelity. Actually, this corresponds to adding a watermark with higher energy into cover data without satisfying the masking constraint, as shown in Fig. 2(d). After subtracting Fig. 2(d) from Fig. 2(a), the resultant residual watermark is that illustrated in Fig. 2(e). By correlating Figs. 2(a) and (e), it is highly possible to reveal the existence of a watermark. Unlike other watermark removal attacks that reduce the quality of the media data, the collusion attack may improve the quality of colluded data. In view of this fact, it is necessary to consider the collusion attack when the robustness of a video watermarking system is evaluated.

3.2 Frame Hash and Video Frame-dependent Watermark

In Sec. 3.1, we found that WEAs are achievable mainly because the hidden watermark behaves like a noise, so anyone can reliably utilize all estimated noise-like watermarks. To disguise this prior knowledge

and hide it from attackers, the key is to reduce the confidence of watermark estimation achieved by WEAs. To this end, we propose the video frame-dependent watermark (VFDW), which must carry information relevant to the video frame itself. Meanwhile, the content-dependent information (called the frame hash herein) must be secured by means of the same secret key K in order for anti-forgery and must be robust against digital processing [15] in order to not affect watermark detection.

Here, the proposed video frame hash extraction procedure is operated in the VLC domain. For each macroblock, a piece of representative, robust information is created. It is defined in each macroblock i as the magnitude relationship between two energies computed from level values:

$$h(i) = \begin{cases} +1, & \text{if } \sum_j |f_j(p_1)| \geq \sum_j |f_j(p_2)|, \\ -1, & \text{otherwise,} \end{cases}$$

where $h(i)$ is an element of a frame hash $\mathbf{FH}$, j is the index that indicates a block belonging to a macroblock i, and $f_j(p_1)$ and $f_j(p_2)$ denote level values at zig-zaged positions p_1 and p_2 in a block j, respectively. The length of an $\mathbf{FH}$ is exactly equal to the number of macroblocks. In addition, the selected level values should be at lower frequencies because level-run pairs located at high-frequency positions are vulnerable to attacks. We say that this feature value $h(\cdot)$ is robust because this magnitude relationship can be mostly preserved under incidental modifications. Since the robustness of the frame hash is beyond the scope of this paper, the reader may refer to [17] for similar robustness analyses.

Next, the frame mash, $\mathbf{FH}$, is merged with the watermark, $\mathbf{W}$, to generate the video frame-dependent watermark ($\mathbf{VFDW}$) as

$$\mathbf{VFDW} = S(\mathbf{W}, \mathbf{FH}), \tag{12}$$

where $S(\cdot, \cdot)$ is a mixing function, which is operated based on a secret key (which will be described in the next section) and is used to prevent attackers from forging the $\mathbf{VFDW}$. The sequence $\mathbf{VFDW}$ is what we will embed into a video frame.

3.3 Properties of the VFDW

Let a video $\mathbf{V}$ be expressed as $\oplus_{i \in \Omega} \mathbf{F}_i$, where all frames $\mathbf{F}_i$ are concatenated to form $\mathbf{V}$ and Ω denotes the set of frame indices. In our video watermarking method, each frame $\mathbf{F}_i$ will be embedded with a video frame-dependent watermark $\mathbf{VFDW}_i$ to form a stego video $\mathbf{V^s}$, i.e.,

$$\mathbf{F^s}_i = \mathbf{F}_i + \mathbf{VFDW}_i, \quad \mathbf{V^s} = \oplus_{i \in \Omega} \mathbf{F^s}_i,$$

where $\mathbf{F^s}_i$ is a stego frame and $\mathbf{VFDW}_i$, similar to Eq. (12), is defined as

$$\mathbf{VFDW}_i = S(\mathbf{W}, \mathbf{FH}_{\mathbf{F}_i}). \tag{13}$$

In Eq. (13), the mixing function $S(\cdot,\cdot)$ is designed for shuffling the frame hash $\mathbf{FH}_{\mathbf{F}_i}$ using the same secret key K, followed by shuffling of the watermark to enhance security for anti-forgery. Specifically, it is expressed as

$$S(\mathbf{W}, \mathbf{FH}_{\mathbf{F}_i})(k) = W(k)PT(\mathbf{FH}_{\mathbf{F}_i}, K)(k),$$

where PT denotes a shuffling function controlled by a secret key K.

The proposed video frame-dependent watermark possesses the characteristics described as below. They are useful for proving resistance to WEAs.

Definition 3 Given two frames $\mathbf{F}_i$ and $\mathbf{F}_j$, their degree of similarity depends on the correlation between $\mathbf{FH}_{\mathbf{F}_i}$ and $\mathbf{FH}_{\mathbf{F}_j}$, i.e., $\delta_{nc}(\mathbf{F}_i, \mathbf{F}_j) = \delta_{nc}(\mathbf{FH}_{\mathbf{F}_i}, \mathbf{FH}_{\mathbf{F}_j})$. Two extreme cases exist: (i) if $\mathbf{F}_i = \mathbf{F}_j$, then $\delta_{nc}(\mathbf{F}_i, \mathbf{F}_j) = 1$; (ii) if $\mathbf{F}_i$ and $\mathbf{F}_j$ are visually dissimilar, then $\delta_{nc}(\mathbf{F}_i, \mathbf{F}_j) \approx 0$.

Definition 4 Given two frames $\mathbf{F}_i$ and $\mathbf{F}_j$, $\delta_{nc}(\mathbf{F}_i, \mathbf{F}_j)$, and their respectively embedded video frame-dependent watermarks $\mathbf{VFDW}_i$ and $\mathbf{VFDW}_j$ that are assumed to be independent and identically distributed (i.i.d.), the following properties can be established: (i) $\delta_{nc}(\mathbf{VFDW}_i, \mathbf{VFDW}_j)$ is linearly proportional to $\delta_{nc}(\mathbf{F}_i, \mathbf{F}_j)$; (ii) $\delta_{nc}(\mathbf{VFDW}_i, \mathbf{VFDW}_j) \leq \delta_{nc}(\mathbf{W}^2)$; (iii) $\delta_{nc}(\mathbf{W}, \mathbf{VFDW}) = 0$. A definition similar to this has been given in [15] for images. It is essential to emphasize that property (i) of Definition 4 contrasts with the one pointed out in [22], and that the novelty of our scheme is that the concept of the content-dependent watermark is employed.

3.4 Resistance to WEA

By means of a collusion attack, the averaging operation is performed on stego frames $\mathbf{F^s}_i$'s of a stego video. From an attacker's perspective, each hidden watermark has to be estimated using a denoising operation [10, 23], so deviation in estimation will inevitably occur. Let $\mathbf{W^e}_i$ be a watermark denoised from $\mathbf{F^s}_i$. In fact, $\mathbf{W^e}_i$ can be modeled as a partial hidden watermark plus a noise component, i.e.,

$$\mathbf{W^e}_i = \alpha_i \mathbf{VFDW}_i + \mathbf{n}_i,$$

where $\mathbf{n}_i$ represents a frame-dependent Gaussian noise with zero mean and α_i denotes the proportion that the watermark has been extracted. Under these circumstances, $1 \geq \alpha_i = \delta_{nc}(\mathbf{W^e}_i, \mathbf{VFDW}_i) > T$ always holds based on the fact that a watermark is a high-frequency signal, which can be efficiently extracted by means of denoising [8, 10, 23]. Let $\mathcal{C}$ ($\subset \Omega$) denote the set of frames used for collusion. After frame collusion is performed, the average of all the estimated watermarks by employing the Central Limit Theorem can be expressed as

$$\bar{\mathbf{W}}^{\mathbf{e}} = \frac{\sqrt{|\mathcal{C}|}}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} \mathbf{W^e}_i = \frac{1}{\sqrt{|\mathcal{C}|}} \sum_{i \in \mathcal{C}} (\alpha_i \mathbf{VFDW}_i + \mathbf{n}_i). \tag{14}$$

Now, a sufficient and necessary condition for resisting a collusion attack will be given in Proposition 2.

Proposition 2 In a collusion attack, an attacker first estimates $\bar{\mathbf{W}}^{\mathbf{e}}$ from a set $\mathcal{C}$ of stego frames. Then, a counterfeit unwatermarked video $\mathbf{V^u}$ is generated from a stego video $\mathbf{V^s} = \oplus_{i\in\Omega}\mathbf{F^s}_i$ according to

$$\mathbf{F^u}_i = \mathbf{F^s}_i - \bar{\mathbf{W}}^{\mathbf{e}}, \qquad \mathbf{V^u} = \oplus_{i\in\Omega}\mathbf{F^u}_i. \tag{15}$$

It is said that the collusion attack fails within a frame $\mathbf{F^u}_k$, $k \in \mathcal{C}$, i.e., $\delta_{nc}(\mathbf{F^u}_k, \mathbf{VFDW}_k) > T$, if and only if $\delta_{nc}(\bar{\mathbf{W}}^e, \mathbf{VFDW}_k) = \frac{\sum_{k\in\mathcal{C}}\alpha_k}{\sqrt{|\mathcal{C}|}} < 1 - T$.

Proof: First of all, we need to derive $\delta_{nc}(\bar{\mathbf{W}}^{\mathbf{e}}, \mathbf{VFDW}_k)$. Making use of Eq. (14) and Proposition 1, we have the following derivation:

$$\begin{aligned}
\delta_{nc}(\bar{\mathbf{W}}^{\mathbf{e}}, \mathbf{VFDW}_k) &= \frac{1}{\sqrt{|\mathcal{C}|}}\delta_{nc}(\sum_{i\in\mathcal{C}}(\alpha_i\mathbf{VFDW}_i + \mathbf{n}_i), \mathbf{VFDW}_k) \\
&= \frac{1}{\sqrt{|\mathcal{C}|}}\sum_{i\in\mathcal{C}}\alpha_i\delta_{nc}(\mathbf{VFDW}_i, \mathbf{VFDW}_k) + \frac{1}{\sqrt{|\mathcal{C}|}}\sum_{i\in\mathcal{C}}\delta_{nc}(\mathbf{n}_i, \mathbf{VFDW}_k) \\
&= \frac{\sum_{k\in\mathcal{C}}\alpha_k}{\sqrt{|\mathcal{C}|}},
\end{aligned} \tag{16}$$

where $\mathbf{VFDW}_k$ represents the content-dependent watermark embedded in $\mathbf{F}_k$. Consequently, given property (ii) of Proposition 1, and Eqs. (15) and (16), we get:

$$\begin{aligned}
\delta_{nc}(\mathbf{F^u}_k, \mathbf{VFDW}_k) > T \quad &\text{iff} \quad \delta_{nc}(\mathbf{F}_k + \mathbf{VFDW}_k - \bar{\mathbf{W}}^{\mathbf{e}}, \mathbf{VFDW}_k) > T \\
&\text{iff} \quad \delta_{nc}(\mathbf{VFDW}_k, \mathbf{VFDW}_k) - \delta_{nc}(\bar{\mathbf{W}}^{\mathbf{e}}, \mathbf{VFDW}_k) > T \\
&\text{iff} \quad \delta_{nc}(\bar{\mathbf{W}}^{\mathbf{e}}, \mathbf{VFDW}_k) = \frac{\sum_{k\in\mathcal{C}}\alpha_k}{\sqrt{|\mathcal{C}|}} < 1 - T.
\end{aligned} \tag{17}$$

Examining the derived result in Proposition 2, we can find from the numerator of $\frac{\sum_{k\in\mathcal{C}}\alpha_k}{\sqrt{|\mathcal{C}|}}$ that the summation function exists because video collusion is generally conducted using a set of similar video frames such that the watermarks extracted from a pair of similar frames can possess a certain positive correlation. However, this characteristic is not guaranteed to hold in images [15]. This difference leads to the fact that video collusion is relatively easier to accomplish than image collusion [§]. Furthermore, resistance to the copy attack can be similarly derived. Please refer to [15] for more details by treating an image as if it were a video frame.

[§]More specifically, we gain an interesting result from the collusion-resilient image watermarking ([15]) and collusion-resilient video watermarking (this paper) methods. In [15], if a collusion attack is carried out in an image watermarking scheme with embedding of multiple redundant watermarks, the anti-collusion performance is proved to be lower bounded by $|\mathcal{C}| = 1$. However, if a collusion attack is carried out in a video watermarking scheme, the anti-collusion performance is almost the same no matter what the size of $|\mathcal{C}|$ is.

4 Experimental Results

A series of experiments was conducted to verify the performance of the proposed method. Three commonly used videos, "Flower-Garden," "Table-Tennis," and "Football," were adopted in our experiments. The number of frames was 375 for both "Flower-Garden" and "Table-Tennis" and 97 for "Football." The frame size of both "Flower-Garden" and "Table-Tennis" was 704×576, and it was 720×486 for "Football." The MPEG-2 codec [18] was used to generate compressed video sequences as the cover sources. The bit-rate of each source video was fixed at 15 Mbits/sec, and the frame-rate was 25 frames/sec. In addition, the length of a GOP was 12, and the GOP structure was "IBBPBBPBBPBB." Because watermarks were concealed in I-frames in this study, we will present the watermark detection results obtained from I frames only even though watermarks also could be found in non-I frames (this will be shown to be true by using I-frame dropping and transcoding attacks). The performance in terms of bit-rate control, fidelity, resistance to numerous incidental and malicious attacks, and real-time detection were examined in our experiments.

4.1 Bit-Rate Control

In order to show the bit-rate control performance of our method, the bit-rates, BR_{cover} and BR_{stego}, generated from MPEG-2 cover video and stego video, respectively, are compared in Table 3. In addition, the ratio, $\frac{BR_{cover}-BR_{stego}}{BR_{cover}}$, of bit-rate reduction achieved by our watermarking method is also given. It can be observed from Table 3 that the bit-rate reduction ratios for these video sequences are all sufficiently small. Numerically, they are on the order of $10^{-4} \sim 10^{-3}$.

4.2 Fidelity of Stego Video Sequences

In order to determine the impact of watermark embedding on the quality of stego video sequences, we compared two PNSR curves that were generated from (i) raw video vs. cover (compressed) video and (ii) raw video vs. stego (compressed+embedded) video. Fig. 3 plots the PSNR curves, where the PSNR values were measured in all frames of the three videos. In Fig. 3 (a), the PSNR values of the first 200 frames vary smoothly while the PSNR values of the remaining frames vary significantly. The reason for these results was that the "Flower-Garden" video had no significant motion at the beginning and started to have significant motion after the 200-th frame. Statistically, an average PSNR decrease of 3.13 dB was generated. Perceptually, no visual degradations could be sensed when the stego video was played normally. Figs. 3 (b) and (c) show similar results that were yielded from the video sequences "Table

Tennis" and "Football." The average PSNR decrease, as shown in Fig. 3 (c), was less than 1 dB. The reason for these results was that some watermark bits were given up to not be embedded in order to enable video re-encoding and decoding to be normally performed.

In Fig. 3, it is not difficult to locate the regions where motion was significant. One thing to note is that the lowest PSNR value was always obtained in an I-frame due to embedding. In addition, the PSNR decrease observed in the B- and P-frames was due to: (i) the referred I-frames were embedded and (ii) drift compensation was not adopted to compensate for the effect of (i) in the proposed watermarking method. Although the overall fidelity of the stego video sequences was more or less degraded, the computational complexity of applying drift compensation was saved. We will show in the following that robustness was not sacrificed even we did not employ drift compensation.

4.3 Resistance to Incidental Video Attacks

To test robustness against different attacks, we used several attacks, including MPEG-2 re-encoding with lower bit-rates (i.e., changing the original bit-rate from $15M$ bps to $6M$, $4M$, and $2M$ bps, respectively), noise addition (the PSNR between a noiseless frame and its noisy version was fixed at 27.05dB), sharpening, frame averaging, and frame rate changing (changed from 25 frames/sec to 24 frames/sec) to verify the performance of our video watermarking algorithm. Since the tested attacks may be applied in normal applications, they are called incidental attacks here. Fig. 4 shows an example of the watermark detection results obtained when Flower-Garden was used as the cover video. In this figure, the horizontal axis indicates the I-frame number, and the vertical axis indicates the correlation value. The correlation values detected in the I-frames of the attacked Flower-Garden video and detected in un-watermarked video I-frames are also provided for the purpose of comparison. In addition, Fig. 4(a) shows the detection results obtained using the proposed method but without using the VFDW, while Fig. 4(b) shows the results obtained using the proposed method with embedding of the VFDW. In both Figs. 4(a) and (b), it is easy to distinguish the attacked watermarked videos from the un-watermarked videos. Furthermore, Fig. 4(b) shows lower correlation values (detected in attacked stego videos) than Fig. 4(a) does. The main reason for these results may have been insufficient randomness of the VFDW (referring to the numbers of 1's and -1's), generated from the contents of video frames (as indicated in Eq. (12)). In particular, when a video frame showed dominant features in either the horizontal or the vertical direction, the resultant video hash was biased. Overall, based on the achieved robustness, our assumption of blind detection accomplished by preserving the intrinsic content of a video, as described in Sec. 2.3.1, is acceptable. In addition, as mentioned previously, the lack of drift compensation did not apparently harm robustness.

In addition to the above attacks, transcoding is also popularly applied to different applications for video transmission over the network. In the following test, transcoding was used to change the GOP structure of stego video sequences. Recall that the original GOP was "IBBPBBPBBPBB" with a length of 12. Three different GOPs, as shown in Table 4, were used in the robustness test. In particular, two of the GOPs had lengths 13 and 19, respectively, such that each of them and 12 (the length of the original GOP) got the greatest common denominator (GCD) 1. Our intention was to (i) make the original I-frames inter-coded in the attacked video; (ii) create new I-frames in the attacked video that were originally inter-coded in the stego video. Fig. 5 shows an example of the changes of the frame types after transcoding. Since the watermarks were embedded into the I-frames, we wanted to evaluate whether the watermarks could still be detected in the new I-frames no matter whether they were B- or P-frames in the stego/un-attacked video. More specifically, we evaluated whether the watermarks, propagated from I-frames to B- and P-frames through transcoding, could still be detected. The correlation values detected from the "I-frames" of the three transcoded video sequences are shown in Fig. 6. It can be observed that (i) the correlation values detected from frames with smaller motions were clearly exceeded the threshold T and (ii) watermarks were harder to detect in frames with larger motions.

4.4 Resistance to Malicious Video Attacks

Here, malicious video attacks refer to those of attackers who may have knowledge about the watermarking algorithm and purposely apply the known knowledge to remove/destroy the hidden watermarks. In the following experiments, we will considered I-frame dropping and WEAs. I-frame dropping was taken into consideration because we assumed that the attackers knew that we had concealed watermarks in I-frames, and that the hidden watermarks would be propagated into B- and P-frames once re-encoding was performed. In addition, watermark estimation attacks, as described and analyzed in Sec. 3.1, are also considered to be malicious.

4.4.1 Resistance to I-Frame Dropping

We started by decoding the watermarked video into a still image sequence, and then those video frames (in the spatial domain) corresponding to the I-frames (in the compressed domain) were removed. After the I-frames had been removed, the remaining frames were re-encoded to yield a new compressed video. Fig. 7 shows three sets of experimental results following the I-frame dropping attack. The horizontal axis denotes the new I-frame numbers of the re-encoded videos, and the vertical axis denotes the correlation values detected in the renewed I-frames. We can see that some detected correlation values were quite

high, which means that the residual watermark (propagated from the original I-frames) could still be detected in the newly assigned I-frames. However, some residual watermarks were almost destroyed. Serious damage to the residual watermark usually happened in those video frames that had significant motion.

In what follows, we will discuss why our method is vulnerable to significant motion. Suppose a watermarked I-frame is at frame t; then frame *t+1* is definitely a B- or P-frame that will refer to the preceding I-frame (frame t). Let $MB_{i,t}$ and $MB_{j,t}$ be the i-th and the j-th macroblock of frame t, respectively. According to our algorithm, the watermark bit $w(i)$ will be embedded in $MB_{i,t}$, and $w(j)$ will be embedded in $MB_{j,t}$. If $MB_{j,t+1}$ must refer to $MB_{i,t}$ during motion estimation, then the content of $MB_{j,t+1}$ is expected to be similar to that of $MB_{i,t}$. The above-mentioned referencing mechanism implies that the watermark bit $w(i)$ of $MB_{i,t}$ of frame t will be propagated to $MB_{j,t+1}$ of frame *t+1*. Under these circumstances, the watermark bit detected from $MB_{j,t+1}$ is $w(i)$, not $w(j)$. However, ideally, the expected watermark bit should be $w(j)$. Therefore, it is clear that when there is any significant motion in a video and the I-frame dropping attack is applied, it is difficult to correctly detect the watermark in the newly assigned I-frames. This explains why the football video produced the worst results (as indicated in Fig. 7(c)).

4.4.2 Resistance to Watermark Estimation Attacks

To study the resistance to WEAs, the proposed video watermarking scheme embedded with a video frame-independent watermark was used and, denoted as Method I. The combination of the proposed VFDW and Method I was denoted as Method II. We wanted to verify the advantage of using VFDW by comparing the performance of Methods I and II when WEAs were imposed.

VFDW Resistance to the Collusion Attack The collusion attack was applied to Method I (without using the VFDW) and Method II (using the VFDW), respectively. The affects of the collusion attack and VFDW were examined from two viewpoints: (s1) the quality of a colluded video; and (s2) watermark detection after performing collusion. Typical results obtained from the Flower-Garden video are depicted in Figs. 8 and 9, respectively.

As for (s1), it can be found in Fig. 8(a) that collusion improved the quality of the colluded video frames in terms of MSE. However, the VFDW could force collusion to undesirably degrade the quality of the colluded video frames, as shown in Fig. 8(b). This experiment demonstrated that the VFDW was efficient in preventing a collusion attack from achieving perfect cover video recovery.

As for (s2), the watermark detection results for colluded video frames are shown in Fig. 9. We can see from the first two curves of Fig. 9 that when video collusion was absent, the detection values obtained from Method II were slightly smaller than those obtained from Method I. On the other hand, the last two curves in Fig. 9 show that when VFDW was not employed (i.e., using Method I), all the watermarks could not be extracted from colluded frames. In addition, once the VFDW was employed in embedding (i.e., using Method II), watermarks could be detected (the 3-rd and 4-th curves of Fig. 9) from all the I-frames no matter what the size of the collusion set $\mathcal{C}$ is. It should be noted that these two curves are very close to each other, which coincide with our result derived in Proposition 2. This experiment verified the resistance of the VFDW to a collusion attack.

In summary, as long as a frame hash is used to construct a watermark, even when a collusion attack is applied, watermarks still can be extracted by owners, and the fidelity of colluded videos cannot be improved. As a result, the merits of VFDW in resisting collusion have been confirmed.

VFDW Resistance to the Copy Attack The copy attack was applied on Method I and Method II, respectively. When the copy attack was performed, one of the videos was first watermarked, and then the watermark was estimated and copied to the other unwatermarked videos to form counterfeit stego videos. By repeating the above procedure, we obtained six counterfeit stego videos in total. The PSNR values (stego video vs. stego+copy attacked video) of the attacked video frames were in the range of $36 \sim 55$dB (no masking was used). The normalized correlations obtained by applying the copy attack to Method I fell within the interval $[0.487 \;\; 0.650]$ (all were sufficiently larger than $T = 0.11$), which indicated the presence of watermarks. However, when VFDW was introduced, these correlations decreased significantly to the interval $[-0.056 \;\; 0.060]$, which indicated the absence of watermarks. The experimental results are consistent with the analytic result indicating that the proposed VFDW is able to deter the detection of copied watermarks.

4.5 Real-Time Detection

As for the real-time detection requirement, the time consumed on the video decoder/watermark detector side in three different situations, including (1) video decoding and re-encoding, (2) video decoding and watermark detection, and (3) video decoding, was compared and is shown in Fig. 10. Our watermarking system was run on a PC with a Pentium-4 2.5GHz CPU under Windows 2000. The two left bars in Fig. 10 show the time needed to finish video decoding plus re-encoding for "Flower-Garden" and "Table-Tennis," respectively. The difference in the time cost is mainly due to the different contents of the video sequences;

therefore the amount of time consumed in motion estimation was different. However, the decoding time required for both video sequences was almost the same. On the other hand, the average watermark detection time was about 0.117 sec/frame. It is obvious that our method (as shown by the third bar in Fig. 10) used much less time than video decoding+re-encoding. Comparing the amount of time used for pure decoding (the bar on the right side of Fig. 10), our method (video decoding+watermark detection) needed nearly the same amount of time. From the compared results, it is reasonable to conclude that our watermark detection scheme can almost be executed in real-time. In addition, uncompressed domain video watermarking is not feasible for real-time application because the time spent in the decoding and re-encoding process is very long.

5 Conclusion

A digital video watermarking system needs to deal with several critical issues that are peculiar to video sequences. In this paper, we have presented a new video watermarking scheme that takes these issues into consideration. These include compressed domain watermarking, real-time detection, bit-rate control, and resistance to video incidental and malicious attacks. In particular, we have provided a watermarking method that can be performed in the VLC domain. We have also designed a video frame-dependent watermark, which is able to resist watermark estimation attacks (WEAs) that have been largely ignored in the literature. Resistance to WEAs is indeed indispensable because they are efficient in defeating a video watermark system while maintaining the visual quality of attacked video sequences. We have conducted extensive experiments to verify the performance of the proposed method.

In this paper, we have not considered resistance to geometrical distortions. As pointed out previously, the embedding of synchronization/repetition patterns is a common way used to tolerate geometrical attacks (only efficient to certain extent). However, the embedded synchronization patterns are easy to remove using the collusion attack. Therefore, we do not think it is sufficient to employ the similar idea to deal with the problem of geometrical distortions. On the contrary, we propose to exploit a mesh [17] or a video object [14, 20] as the basic embedding unit, and to combine it with its content-dependent information to resist both the geometrical distortion and collusion attacks. Of course, the robustness of mesh [17] or video object extraction plays an important role. We are currently studying these topics.

Acknowledgment: This paper was supported under NSC grants 91-2213-E-001-037 and 92-2422-H-001-004.

References

[1] 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 Technology*, Vol. 13, No. 8, pp. 787-800, 2003.

[2] S. Arena, M. Caramma, and R. Lancini, "Digital Watermarking Applied to MPEG-2 Coded Video Sequences Exploiting Space and Frequency Masking," *Proc. IEEE Int. Conf. on Image Processing*, 2000.

[3] S. Baudry and P. Nguyen and H. Maita, "Channel coding in video watermarking: use of soft decoding to improve the watermark retrieval," *Proc. IEEE Int. Conf. on Image Processing*, 2000.

[4] I. J. Cox, M. L. Miller, and J. A. Bloom, "Digital Watermarking," *Morgan Kaufmann Publishers*, 2002.

[5] J. Dittmann, M. Stabenau, and R. Steinmetz, "Robust MPEG Video Watermarking Technologies," *Proc. ACM Multimedia*, Bristol, UK, 1998.

[6] F. Hartung and B. Girod, "Watermarking of Uncompressed and Compressed Video," *Signal Processing*, Vol. 66, No. 3, pp. 283-302, 1998.

[7] F. Hartung and M. Kutter, "Multimedia Watermarking Techniques," *Proceedings of the IEEE*, Vol. 87, pp. 1079-1107, 1999.

[8] T. Kalker, G. Depovere, J. Haitsma, and M. Maes, "A Video Watermarking System for Broadcast Monitoring," *Proc. of the SPIE*, Vol. 3657, pp. 103-112, 1999.

[9] D. R. Kim and S. H. Park, "A Robust Video Watermarking Method," *Proc. IEEE Int. Conf. on Multimedia and Expo*, 2000.

[10] M. Kutter, S. Voloshynovskiy, and A. Herrigel, "The Watermark Copy Attack", *Proc. SPIE: Security and Watermarking of Multimedia Contents II*, Vol. 3971, 2000.

[11] G. C. Langelaar, R. L. Lagendijk, and J. Biemond, "Real-Time Labeling of MPEG-2 Compressed Video," *Journal of Visual Communication and Image Representation*, Vol. 9, No. 4, pp. 256-270, 1998.

[12] G. C. Langelaar and R. L. Lagendijk, "Optimal Differential Energy Watermarking of DCT encoded Images and Videos," *IEEE Trans. on Image Processing*, Vol. 10, No. 1, pp. 148-158, 2001.

[13] J. Linnartz and J. C. Talstra, "MPEG PTY-Marks: Cheap Detection of embedded Copyright Data in DVD-Video," *ESORICS98.*, 1998.

[14] C. S. Lu and H. Y. Mark Liao, "Video Object-based Watermarking: A Rotation and Flipping Resilient Scheme," *Proc. IEEE Int. Conf. on Image Processing*, Greece, Vol. 2, 2001.

[15] C. S. Lu and C. Y. Hsu, "Content-dependent Anti-Disclosure Image Watermark", *Proc. Int. Workshop on Digital Watermarking*, LNCS 2939, Seoul, Korea, 2003.

[16] C. S. Lu, J. R. Chen, and K. C. Fan, "Resistance of Content-dependent Video Watermarking to Watermark-Estimation Attacks," *Proc. IEEE Int. Conf. on Communications*, France, 2004.

[17] C. S. Lu, C. Y. Hsu, S. W. Sun, and P. C. Chang, "Robust Mesh-based Hashing for Copy Detection and Tracing of Images," submitted to *IEEE Int. Conf. on Multimedia and Expo*, Taipei, Taiwan, 2004.

[18] *ftp://ftp.mpegtv.com/pub/mpeg/mssg/mpeg2v12.zip.*

[19] F. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information Hiding: A Survey," *Proc. of the IEEE*, Vol. 87, pp. 1062-1078, 1999.

[20] A. Piva, R. Caldelli, and A. D. Rosa, "A DWT-based Object Watermarking System for MPEG-4 Video Streams," *Proc. IEEE Int. Conf. on Image Processing*, Vol. III, pp. 5-8, 2000.

[21] K. Su, D. Kundur, D. Hatzinakos, "Statistical Invisibility for Collusion-resistant Digital Video Watermarking," to appear in *IEEE Trans. on Multimedia.*

[22] M. D. Swanson, B. Zhu, and A. H. Tewfik, "Multiresolution Scene-based Video Watermarking Using Perceptual Models," *IEEE Journal on Selected Area in Communications*, Vol. 16, No. 4, pp. 540-550, 1998.

[23] S. Voloshynovskiy, S. Pereira, A. Herrigel, N. Baumgartner, and T. Pun, "Generalized Watermarking Attack Based on Watermark Estimation and Perceptual Remodulation", *SPIE: Security and Watermarking of Multimedia Contents II*, Vol. 3971, 2000.

[24] S. Voloshynovskiy, F. Deguillaume, S. Pereira, and T. Pun, "Optimal Adaptive Diversity Watermarking with Channel State Estimation," *Proc. SPIE: Security and Watermarking of Multimedia Contents III*, Vol. 4314, USA, 2001.

Table 1: **Video Watermarking Methods vs. Video Characteristics**

Video characteristic	[22]	[8]	[21]	[6]	[11]	[12]	[1]
Compressed domain watermarking	N	N	N	Y(DCT)	Y(VLC)	Y(DCT)	Y(DCT)
Real-time detection	N	Y	N	Y	Y	Y	Y
Bit-rate (nearly) unchanged	N	N	N	N	Y	N	Y
Drift compensation	N	N	N	Y	N	N	Y
Low bit-rate embedding	N	N	N	N	N	N	Y
Resistance to Collusion	Y	N	Y	N	N	N	N
Resistance to Copy attack	N	N	N	N	N	N	N

Table 2: **Variable Length Codeword (VLC) Table (s denotes the sign bit)**

$(run, level)$	Variable length code	Bit length
(0,1)	11**s**	3
(0,2)	0100 **s**	5
(0,3)	0010 1**s**	6
(0,4)	0000 110**s**	8
(0,5)	0010 0110 **s**	9
(0,6)	0010 0001 **s**	9
:	:	:
(1,1)	011**s**	4
(1,2)	0001 10**s**	7
(1,3)	0010 0101**s**	9
:	:	:

Table 3: **Results for Bit-Rate Control: the ratios of the bit-rate decrease obtained from three video sequences are within the order of** $[10^{-4}\ \ 10^{-3}]$

Video	BR_{cover} (bytes)	BR_{stego} (bytes)	$\frac{BR_{cover}-BR_{stego}}{BR_{cover}}$
Flower-Garden	$28,120,582$	$28,109,649$	$3.89e^{-4}$
Table-Tennis	$28,118,904$	$28,089,882$	$1.03e^{-3}$
Football	$7,276,845$	$7,271,528$	$7.31e^{-4}$

Table 4: **Transcoding Parameters**

GOP's structure	IBBBBBBBBBBBB	IBBPBBPBBPBBPBB	IPPPPPPPPPPPPPPPPP
GOP's length	13	15	19

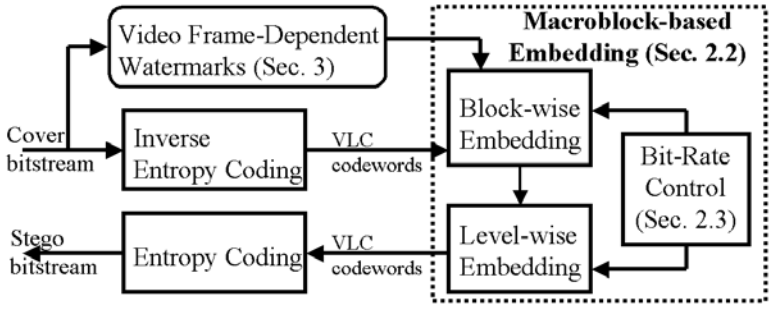


Figure 1: Block diagram of the proposed watermark embedding process.

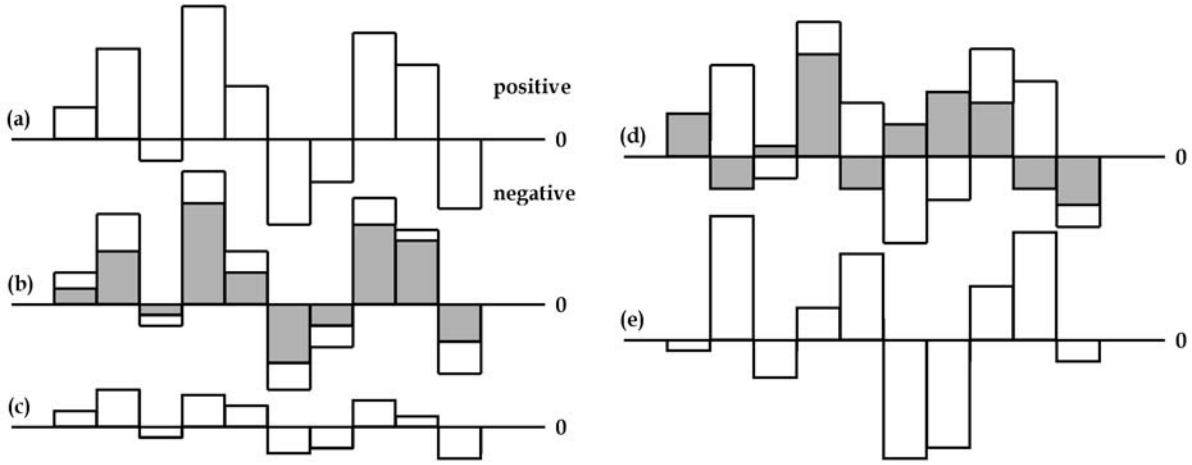


Figure 2: Watermark estimation/removal illustrated with energy variations: (a) original embedded watermark with each white bar indicating the energy of each watermark value; (b) gray bars show the energies of an estimated watermark with all the signs being the same as in the original (a); (c) the residual watermark obtained after removing the estimated watermark (b); (d) the energies of an estimated watermark with most of the signs being opposite to those in (a); (e) the residual watermark derived from (d). In the above examples, sufficiently large correlations between (a) and (c), and between (a) and (e) exist, indicating the presence of a watermark.

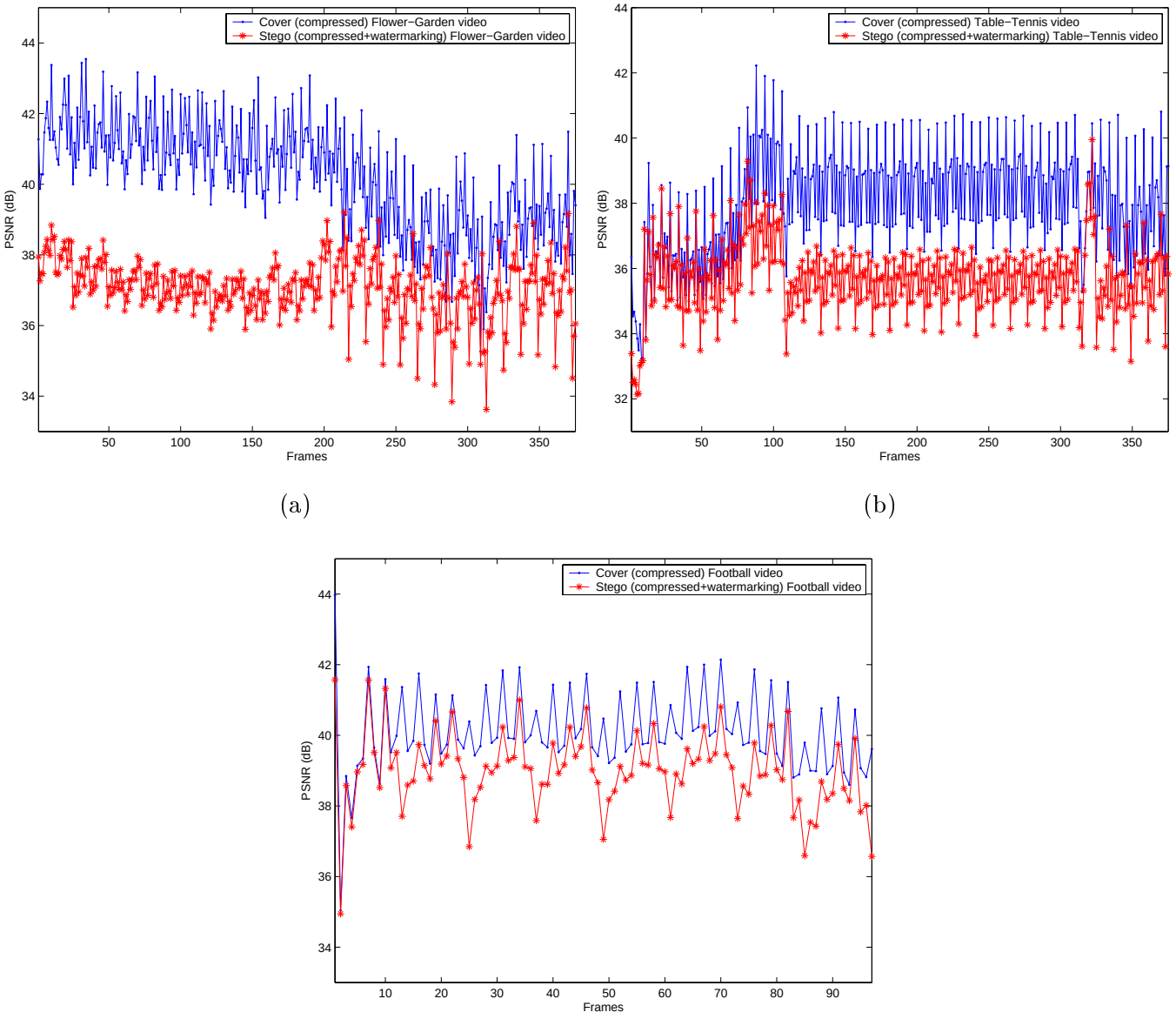
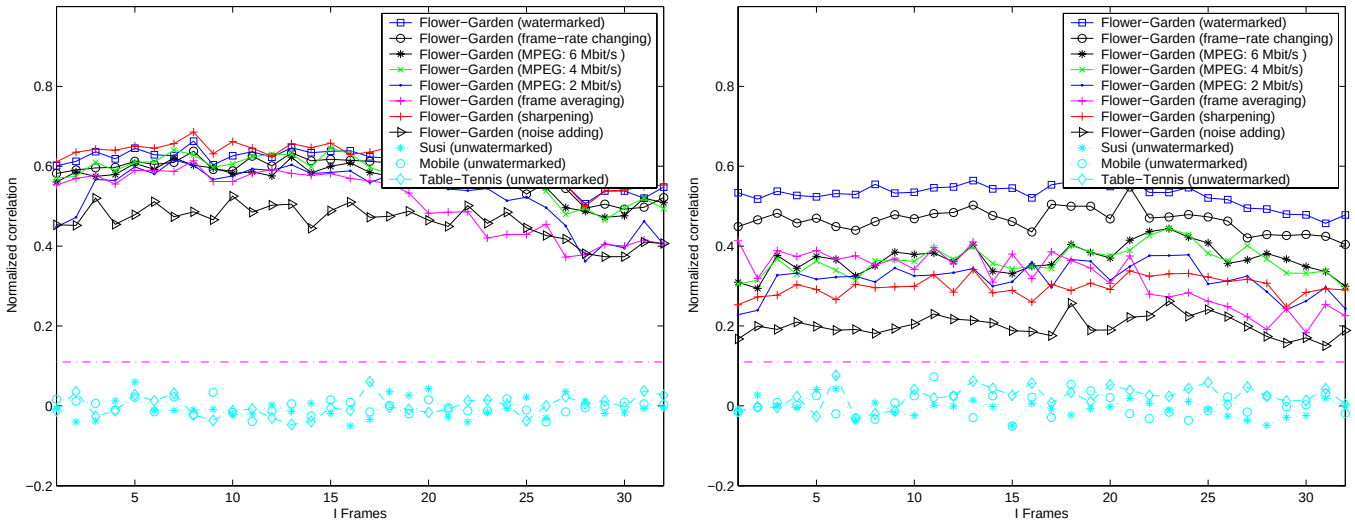


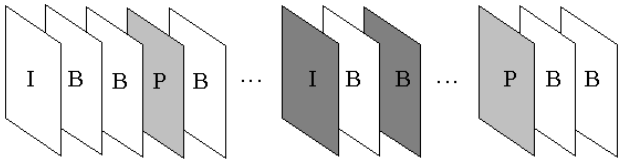
Figure 3: The PSNR values measured in different frames of three videos: (a) Flower-Garden; (b) Table-Tennis; (c) Football. The PSNR decrease (in dB) for each video is indicated statistically in terms of the mean/variance: (a) 3.13/1.35; (b) 2.32/1.26; (c) 1.11/0.69.



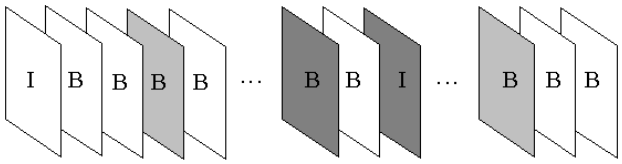
(a) Our method without using the VFDW

(b) Our method using the VFDW

Figure 4: The correlation values detected from the attacked Flower-Garden video sequences and unwatermarked videos. (a) shows the results obtained using our method but without using the VFDW, while (b) shows the results obtained using our method by embedding the VFDW. The dashdot line indicates the threshold $T = 0.11$.

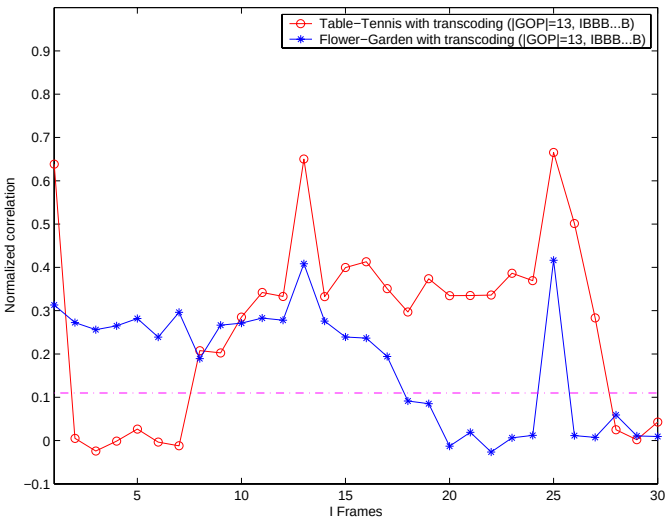


(a) Original GOP structure: IBBPBBPBBPBB

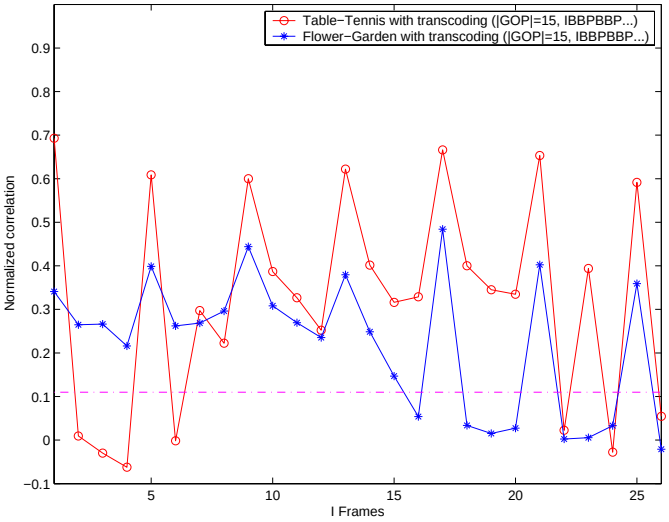


(b) New GOP structure: IBBBBBBBBBBB

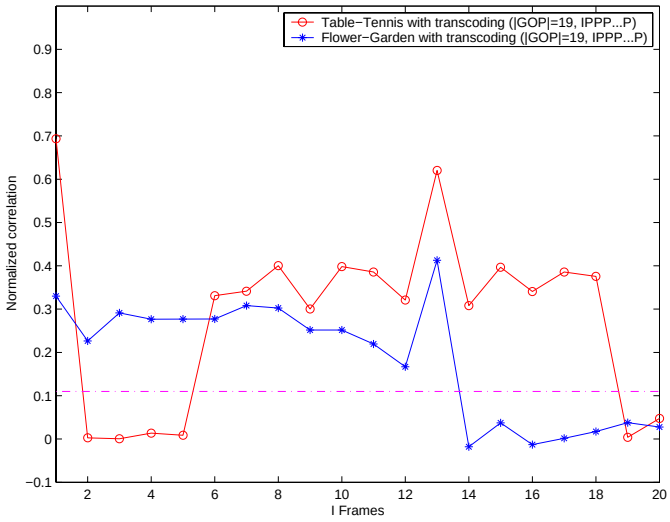
Figure 5: Change of the Group of Picture (GOP): (a) a video encoded with an original GOP; (b) a video encoded with a new GOP under transcoding. Light gray shading indicates the frame types that are changed from P to B, while dark gray shading indicates the frame types that are changed from I to non-I or from non-I to I.



(a) GOP Type I

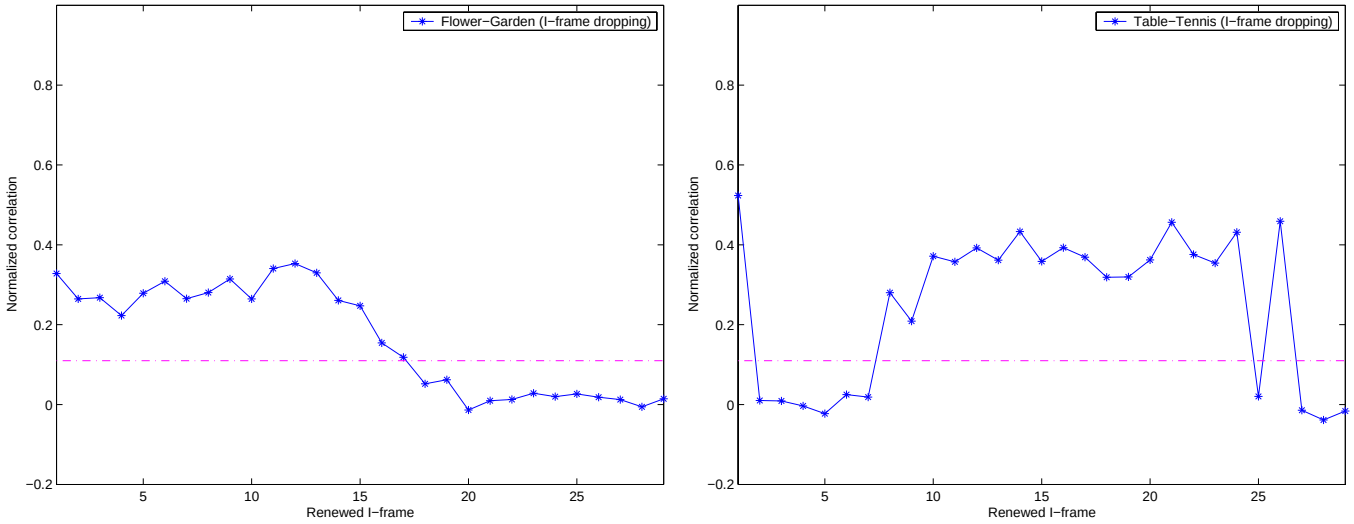


(b) GOP Type II



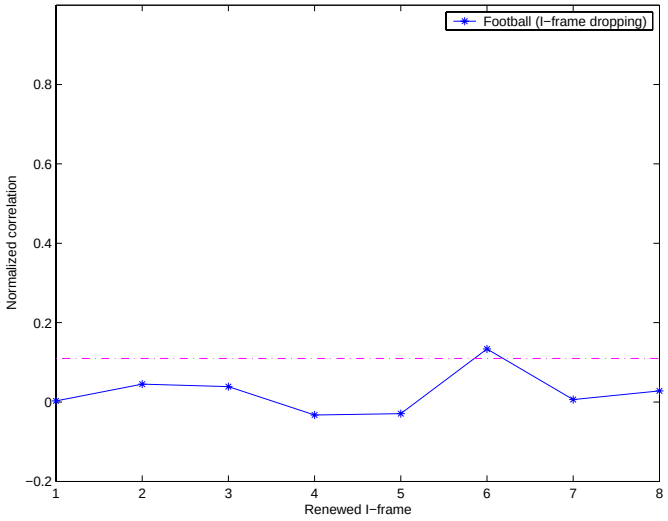
(c) GOP Type III

Figure 6: The correlation values detected from transcoded videos using the GOP parameters described in Table 4. The dashdot line indicates the threshold $T = 0.11$.



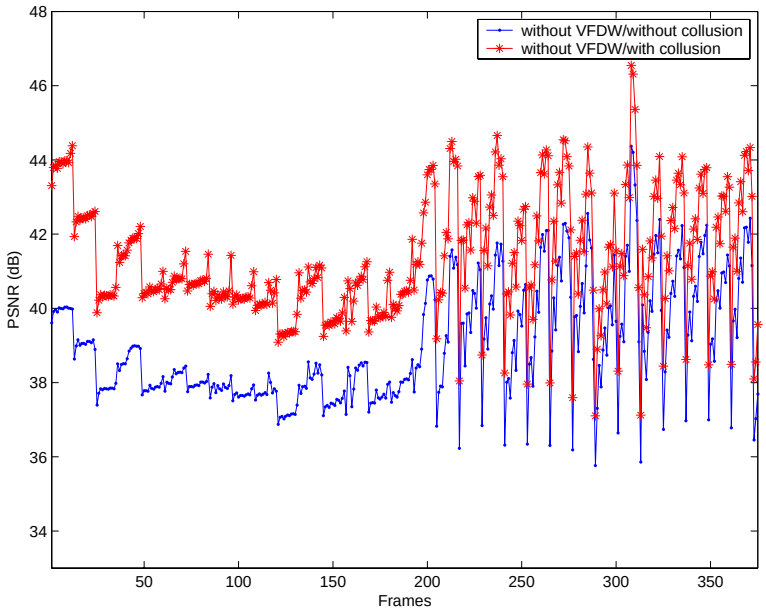
(a)

(b)

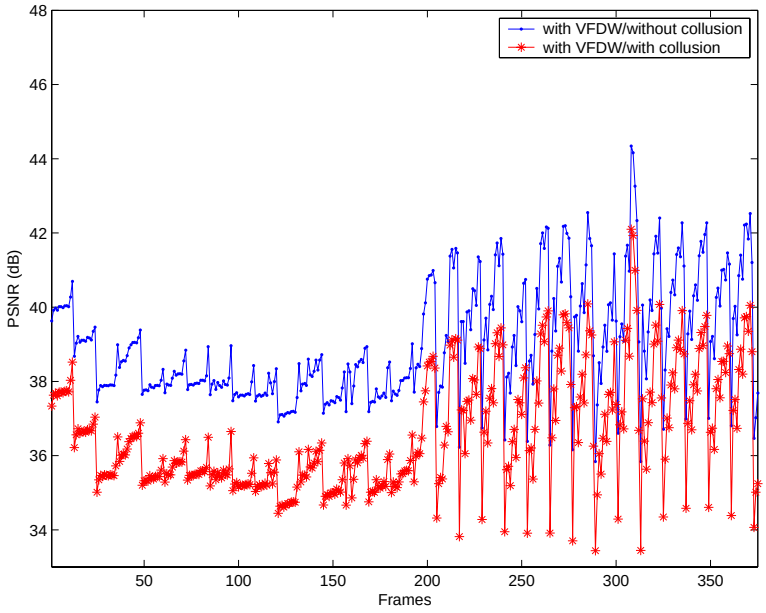


(c)

Figure 7: The correlation values detected after the I-frame dropping attack was applied to the (a) Flower-Garden, (b) Table-Tennis, and (c) Football, respectively. The dashdot line indicates the threshold $T = 0.11$.



(a) Method I (without using VFDW)



(b) Method II (using VFDW)

Figure 8: Quality of a colluded Flower-Garden video: (a) the PSNR values of the colluded frames (top) are higher than those of the stego frames; (b) when VFDW was applied, the PSNR values of the colluded frames (bottom) became lower than those of the stego frames. This experiment reveals that a collusion attack will fail to improve the fidelity of a colluded video when VFDW is applied.

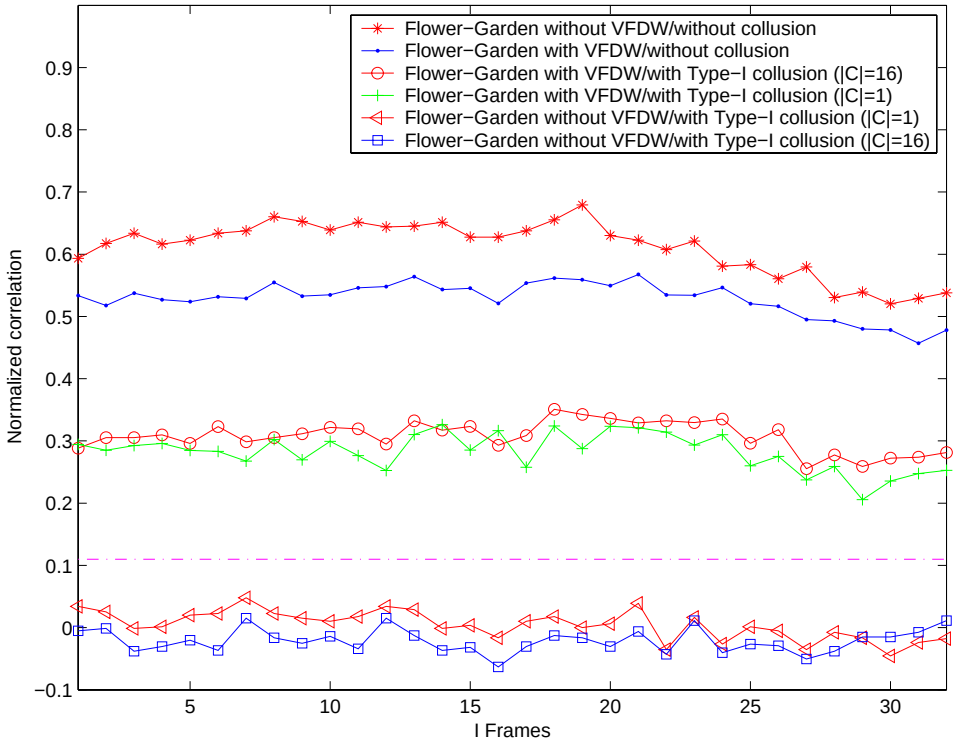


Figure 9: Watermark detection under collusion (the dashdot line indicates the threshold $T = 0.11$). Comparing the detection curves obtained using different $|\mathcal{C}|$'s reveals that the anti-collusion capability of our watermarking method is not affected by the size of a collusion set (see the 3-rd and 4-th curves). If VFDW is not used, collusion indeed provides effective watermark removal (see the last two curves). These results are exactly consistent with Proposition 2.

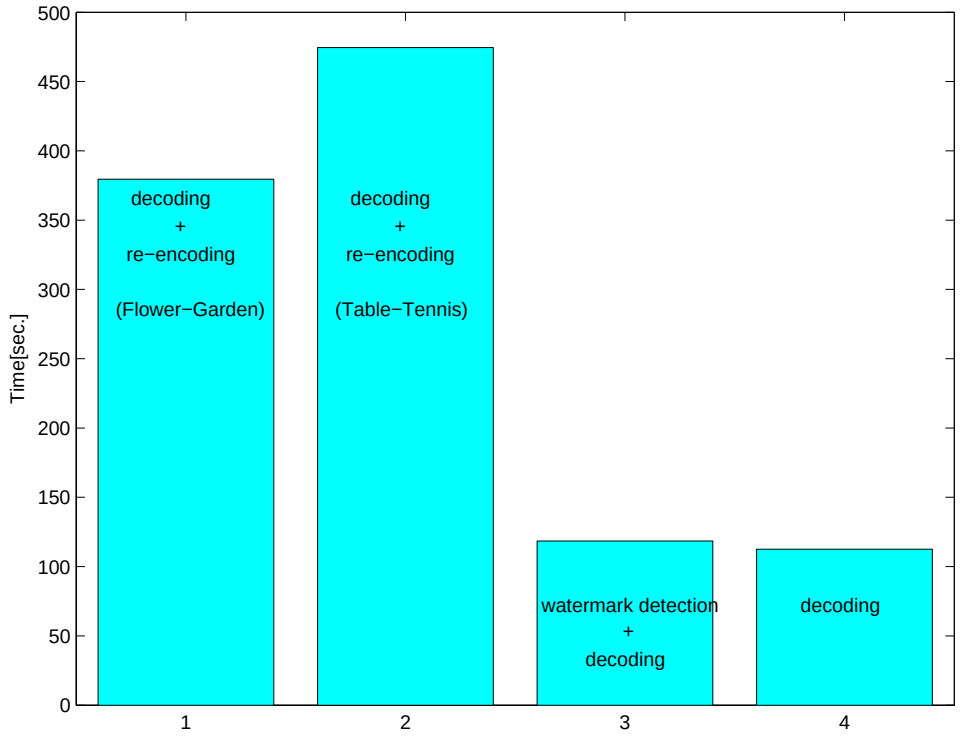


Figure 10: Comparison of the time consumed by (1) Flower-Garden video decoding+re-encoding; (2) Table-Tennis video decoding+re-encoding; (3) video decoding+our watermark detection (not optimized for speed), and (4) video decoding, respectively. Note that the amount of time needed to re-encode Flower-Garden and Table-Tennis were different. However, the amount of time used in watermark detection+decoding and decoding for both Flower-Garden and Table-Tennis were almost the same.