Tampering Detection in Compressed Digital Video Using Watermarking

Mehdi Fallahpour, Shervin Shirmohammadi, *Senior Member, IEEE,*
Mehdi Semsarzadeh, and Jiying Zhao, *Member, IEEE*

***Abstract*— This paper presents a method to detect video tampering and distinguish it from common video processing operations, such as recompression, noise, and brightness increase, using a practical watermarking scheme for real-time authentication of digital video. In our method, the watermark signals represent the macroblock's and frame's indices, and are embedded into the nonzero quantized discrete cosine transform value of blocks, mostly the last nonzero values, enabling our method to detect spatial, temporal, and spatiotemporal tampering. Our method can be easily configured to adjust transparency, robustness, and capacity of the system according to the specific application at hand. In addition, our method takes advantage of content-based cryptography and increases the security of the system. While our method can be applied to any modern video codec, including the recently released high-efficiency video coding standard, we have implemented and evaluated it using the H.264/AVC codec, and we have shown that compared with the existing similar methods, which also embed extra bits inside video frames, our method causes significantly smaller video distortion, leading to a PSNR degradation of about 0.88 dB and structural similarity index decrease of 0.0090 with only 0.05% increase in bitrate, and with the bit correct rate of 0.71 to 0.88 after H.264/AVC recompression.**

***Index Terms*— Video authentication, video tampering detection, video watermarking.**

I. Introduction

THE FAST growth of the Internet, sudden production of low-cost and reliable storage devices, digital media production, and editing technologies have led to widespread forgeries and unauthorized sharing of digital media. Among these media, video is becoming increasingly important in a wide range of applications, such as video surveillance, video broadcast, DVDs, video conferencing, and video-on-demand applications, where authenticity and integrity of the video data is crucial. In surveillance applications [1]–[3], significant investments have been made in infrastructure, such as video cameras and networks installed in public facilities on a wide scale. However, current video editing software can be used to tamper with such video, making them unreliable and defeating the purpose of such applications at the first place. Without authentication [4], [5], a video viewer (or a consumer) cannot verify that the video being viewed is really the original one that was transmitted by a producer. There may be some eavesdroppers who modify the video content intentionally to harm the interests of either or both the producer and the consumer. There is therefore a need to not only detect such tamperings, but also to distinguish them from common video processing operations, such as compression. As a side effect, video authentication can also be used for advertisement monitoring, where a company can automatically identify, in real time, whether or not a specific TV or Internet channel has cut a few frames of the company's advertisement to gain more time and money. Considering these different applications, authentication systems are becoming popular to ensure the integrity of video content.

A well-known solution to the above problems are watermarking, which hides important information in the media. A well-designed watermarking system must provide three main features: 1) transparency; 2) robustness; and 3) capacity. Transparency means that the marked signal should be perceptually equivalent to the original signal, robustness refers to a reliable extraction of the watermark even if the marked signal is degraded, and capacity is a measure of how much information can be embedded into the media. While the original motivation behind watermarking was copyright protection, watermarking can also be used for verifying the authenticity and integrity of the video by embedding the watermark information behind a cover. The embedded watermark can then be detected or extracted from the cover video used for verification.

In contrast to robust watermarking, which is designed for copyright protection, fragile watermarking [6] has been designed for tamper detection. An attacker's goal in tampering is to change the watermarked media while keeping the watermark itself untouched, so as to trick the receiver into believing that the tampered media is authentic and has integrity. While fragile watermarking can protect against such an attack, it is highly sensitive to modifications, making it difficult to distinguish malicious tampering from some common video processing operations, such as recompression. To exploit the advantages of both the robust and the fragile schemes, semifragile watermarking [7]–[9] has been proposed to tolerate common processing, such as recompression, and at the same time detect malicious tampering.

In this paper, we introduce a watermarking scheme that can be used to detect malicious tampering. Our scheme can be used

Manuscript received June 14, 2013; revised December 2, 2013; accepted December 30, 2013. Date of publication January 21, 2014; date of current version April 3, 2014. The Associate Editor coordinating the review process was Dr. Amitava Chatterjee.

The authors are with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON K1N 6N5, Canada (e-mail: mfallah@site.uottawa.ca; shervin@eecs.uottawa.ca; msemsarz@uottawa.ca; jyzhao@eecs.uottawa.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIM.2014.2299371

in any modern video codec, and can survive compression by advanced codecs, such as H.264/AVC, whereas many existing tampering detection schemes are fragile against H.264/AVC compression. In our proposed scheme, macroblocks' (MBs') and frames' indices are embedded into the last nonzero (LNZ) quantized discrete cosine transform (QDCT) value of the blocks. Using high frequency levels leads us to assure transparency to the human visual system. Compared with the existing H.264/AVC watermarking schemes, our solution has five benefits: 1) it addresses both spatial and temporal domains, which leads to detecting various malicious changes in spatial and time domains; 2) it is faster and with lower complexity compared to existing algorithms, making it practical and suitable for real-time applications; 3) its implementation is simple and requires minor changes in the codec; 4) it provides high transparency and high capacity; and 5) the imposed bitrate increase for the compressed H.264/AVC bitstream is practically near zero (around 0.05%), unlike existing schemes that increase the bitrate significantly, usually in the 3%–15% range.

A preliminary report of our method was presented in [10]. In this paper, we have expanded our design and now embed the watermark signals not only in the LNZ values as we did in [10], but also in other nonzero quantized discrete cosine transform (DCT) value of blocks, including those in the middle or even low frequencies. This increases the security of our scheme and makes it more difficult for an intruder to undetectably tamper with the video, although naturally it also increases distortion. We have also improved the bitrate overhead from 3.6% in [10] to only 0.05% in this paper. Finally, in this paper, we are presenting comprehensive evaluations of our algorithm and tempering detection, significantly adding to or expanding the evaluations reported in [10].

The rest of this paper is organized as follows: in Section II, a presentation of the related work is given, while Section III describes the requirements of a watermarking system from an authentication perspective, as well as our proposed scheme. Section IV presents our experimental results and analysis, before concluding this paper in Section V.

II. Related Work

There has been much research activity in using video watermarking for authentication and tampering detection. For example, [1] suggests an authentication method based on chaotic semifragile watermarking. The timing information of video frames is modulated into the parameters of a chaotic system. Then, the output chaotic stream is used as watermark and embedded into the block-based DCT domain of video frames. Their timing information for each frame is modulated into the parameters of the chaotic system. A mismatch between the extracted and the observed timing information is able to reveal temporal tampering. Unfortunately, [1] is in the uncompressed domain and cannot be applied directly to H.264/AVC. Since almost all digital video products today are distributed and stored in the compressed format, compressed-domain video processing [11]–[13] is very attractive. As such, various watermarking methods tailored to MPEG2 and MPEG4 [13]–[16] as well as H.264/AVC [17] have been proposed. Reference [18] can detect cut-and-splice or cut-insert-splice operation by embedding a watermark with a strong timing content. Reference [19] employs error-correcting code (ECC) to propose a secure and robust authentication scheme, which is insensitive to incidental distortions while sensitive to intentional distortions, such as frame alterations and insertion. A video content authentication algorithm for MPEG-2 was described in [20] where the watermark bits were generated according to the image features of I-frame and embedded into the low-frequency DCT coefficients. Reference [21] is a novel video authentication method considering the moving objects. However, this scheme does not consider the effects of H.264/AVC compression in the results. Reference [22] presents a method for applying a watermark directly to an entropy coded H.264/AVC stream by building an embedding table, while [23] exploits the intrapulse code modulation (IPCM), which has two disadvantages: 1) the rareness of the IPCM MBs during encoding and 2) the low efficiency of the IPCM mode in terms of compression. In [24], a fragile video watermarking scheme to authenticate the H.264/AVC video content is presented in which watermark information is embedded in motion vectors. These fragile watermarking methods [22]–[24] are designed for authentication; however, due to their high sensitivity to modifications, it is difficult to distinguish malicious tampering from common video processing. To apply data hiding to content authentication, a semifragile watermarking technique could be considered to tolerate certain kinds of processing [25], such as recompression, and at the same time detect malicious tampering manipulations. Reference [26] suggests a content-based MPEG video authentication system, which is robust to typical video transcoding approaches, namely frame resizing, frame dropping, and requantization. Finally, [27] embeds the edge map of each frame in the video stream. During the detection process, if the video content has been modified, there will be a mismatch between the extracted edge map from the modified video and the watermarked edge map.

Most of the current watermarking schemes focus on the frequency domain rather than the spatial domain because the characteristics of the video in the frequency domain are more robust, invisible, and stable. The common well-known frequency domain methods are the DCT, discrete Fourier transform, and discrete wavelet transform [28]. Among these, DCT is more popular and beneficial [29]–[31] since most of the encoding schemes, including high-efficiency video coding (HEVC) and H.264/AVC, use it in their encoding process. Zhang *et al.* [32] embed 2-D eight-bit grayscale image watermarks in the compressed domain for copyright protection. After preprocessing, one bit of the watermark is embedded into the sign of one middle-frequency coefficient in the diagonal position in a 4×4 DCT block. In [33], we see a low complexity video watermarking scheme in the H.264/AVC compressed domain that avoids full decoding and reencoding in both embedding and extracting phases. Reference [34] embeds watermarks in the H.264/AVC video by modifying the quantized dc coefficients of the luma residual blocks. To increase robustness while maintaining the perceptual quality of the video, a texture-masking-based perceptual model is used to adaptively choose the watermark strength for each block. Finally, [35] proposes a watermarking scheme for H.264/AVC using a spatiotemporal just-noticeable difference model, which is based on 4×4 DCT blocks.

To the best of our knowledge, compared with the above works, our approach is faster, more transparent, and more robust against recompression. In addition, due to very low complexity, simplicity, ease of implementation, low overhead, and efficiency of our approach in terms of capacity, transparency, and security, it works as an excellent solution for real-time video authentication applications.

III. Proposed Watermarking Scheme

While the proposed scheme can be used for all video watermarking applications, such as copyright protection, in this paper, we focus on authentication and tampering detection. Each application, including authentication, has its own requirements. With the requirements of an authentication application, here we design a semifragile watermarking method. The summary of our design is as follows.

Most of the traditional watermarking schemes are not robust against compression, especially HEVC or H.264/AVC compression, and after compression, the secret embedded information is not detectable. In contrast, our proposed scheme takes advantage of the compression standard to embed and extract secret bits. After performing DCT and the quantization phases, some 4×4 blocks of each 16×16 MB are selected for embedding. With the number of secret bits which will be embedded into an MB, the number of selected blocks is chosen. In each MB, the blocks that have larger LNZ level position are selected, i.e., blocks that have the highest high-frequency sample. Choosing high-frequency QDCT values imposes lower modification distortion. In each selected block, a single secret bit is embedded. If the corresponding secret bit is zero, the sum of all levels should be even. If it is odd, the LNZ level is incremented or decremented by one. If the secret bit is one, the sum should be odd. However, if the sum is even, the LNZ level should be incremented or decremented by one. For increased robustness, we also use some other nonzero levels, though at a tradeoff with increased distortion. The experimental results prove that the proposed scheme is transparent, high capacity, and robust against common signal processing operations. Selecting blocks based on the frequency samples leads to an adaptive video watermarking system with its capacity, transparency, and robustness adjusted easily.

In our suggested method, the index of each MB is embedded inside of it; also, the index of each frame is embedded into the current frame. Thus, this scheme provides a good solution for both spatial and temporal tampering. H.264/AVC recompression, noise, filtering, and other spatial changes cause some errors in tamper detection. However, in our scheme, analyzing the extraction error will distinguish malicious attacks from common processing, such as compression, as will be shown in Section IV. In addition, extracted frames' indices help us to recognize any frame manipulation, such as adding extra frames, reordering, dropping, or replacement of video frames.

With the above summary in mind, let us now take a closer look at the details of our design, starting with some definitions and explanations of related concepts.

A. Tampering

Video tampering schemes can be classified into spatial tampering, temporal tampering, or combination of them. Spatial tampering, also called intraframe tampering, refers to changing the image frame, such as cropping and replacement, content adding and removal. Temporal tampering, also named interframe tampering, is the changes made in the time domain, such as adding extra frames, reordering the sequence of frames, dropping, and replacing frames. Due to temporal redundancy in video data, it is possible to perform temporal tampering without imposing visual distortion and semantic alteration. Thus, having an authentication system for temporal tampering detection is inevitable.

B. Transparency, Capacity, and Robustness

The watermarking process should not introduce any perceptible artifacts into the original contents. Ideally, there must be no perceptible difference between the watermarked and the original digital contents, i.e., the watermark data should be transparent to the user. Apart from transparency, capacity and robustness are two other fundamental properties of video watermarking. Capacity is defined as the number of bits embedded in one second of the video. For robustness, the watermark should be extractable after various intentional or unintentional attacks. These attacks may include additive noise, resizing, low-pass filtering, and any other attack, which may remove the watermark or confuse the watermark extraction system. The tradeoff between capacity, transparency, and robustness is the main challenge for video watermarking applications, i.e., in an ideal case, we would demand a very transparent, robust, and high-capacity scheme. However, in practice, obtaining all these properties at the same time is extremely difficult or even impossible. Thus, depending on the requirements of the particular application at hand, a tradeoff between these properties must be attained.

Considering this tradeoff, the following types of watermarking schemes lead to different capacity, transparency, and robustness.

1) *Fragile:* Very high capacity and transparency can be achieved.
2) *Semifragile:* Robustness against compression and common signal processing operations is obtained. In this case, it is accepted that more distortion is caused compared with fragile watermarking. The main application of this category is authentication, which is the main target of this paper.
3) *Robust:* Robustness against many attacks with a wide range of changes is achieved. This is more complicated than the previous two types, since we need robustness against most of the attacks. Thus, according to the trade-off between capacity, transparency, and robustness,

TABLE I

RATE OF MODE AND GROUP CHANGES AFTER REENCODING CONSIDERING NUMBER OF NONZERO LEVELS. DRAWN FROM 100 FRAMES OF *Container*, *Foreman*, *Mobile*, *News*, AND *Tennis*. COMPRESSED USING QP = 24

Number of Nonzero levels	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Mode change rate	0.27	0.26	0.22	0.17	0.13	0.09	0.07	0.05	0.04	0.04	0.03	0.03	0.03	0.02	0.02	0.01	0.01
Group rate change	0.09	0.09	0.07	0.06	0.04	0.03	0.02	0.02	0.01	0.01	0.01	0.01	0.005	0.005	0.00	0.00	0.00

a sacrifice in capacity and transparency is made. The main application of this category is copyright protection.

Our proposed scheme takes advantage of the codec to embed the secret information so that watermarking can be detected at the decoder side, i.e., embedding the watermark is a part of the video encoding process. Using the encoder solves the problem of robustness against compression and also leads to very low complexity since the proposed method uses DCT blocks, which are already computed by all modern video encoders, including HEVC and H.264/AVC. In our proposed method, QDCT coefficients (also known as levels) of some blocks are manipulated to embed the watermark signals, as described next.

C. Embedding

Embedding in the low frequency levels of 4×4 blocks, which carry perceptually important information, results in obvious quality distortion in the watermarked video. Therefore, in our proposed scheme, we mostly use the latest nonzero QDCT coefficients of 4×4 blocks, named LNZ levels, which are in the high or midfrequency bands to embed the watermark. For increased robustness, we also use some other nonzero levels, though at a tradeoff with increased distortion.

In each 16×16 MB, we embed k ($k < 16$) bits. In fact, k 4×4 blocks among the 16 blocks of each MB are selected for embedding, while a single bit is embedded in each selected block. If all levels in a block are zero and there is no LNZ, as can happen in high QP values, we cannot embed inside that block. Thus, before choosing k, the number of blocks that have LNZ should be considered. The embedding process is performed after the quantization phase by following the steps below, which lead to embedding k bits in the current MB.

1) For each 4×4 block within the current MB, find the position of the LNZ level.
2) Select k blocks that have LNZ levels in higher positions, call them block$_i$ ($i = 1$ to k). In other words, we select blocks that have high frequency levels. For example, a block with its LNZ level located at position 12 has priority over another block with its LNZ level at position 3.
3) To improve the security, the authentication code As, is encrypted by a key called C, to form the watermark signal W

$$W = E(C, As)$$

where E is the encryption operation, and As is the binary MB number with a length of k bits. In the experimental results, since we have used QCIF clips (176×144), there are 99 MBs in each frame that need to be marked. To generate 99 numbers, seven bits are required (k $\geq$ 7).
4) In each MB, do step 5 for each selected block$_i$. If a block is not selected for embedding, it should be left as is.
5) For each selected block$_i$, compute the sum, S_i, of all levels within the block and modify its LNZ level (L_i) based on S_i and w_i, as follows:

$$L_i = \begin{cases} L_i + 1 & \text{if } S_i \text{ is odd, } w_i \text{ is zero and } L_i \neq -1 \\ L_i - 1 & \text{if } S_i \text{ is odd, } w_i \text{ is zero and } L_i = -1 \\ L_i & \text{if } S_i \text{ is odd, } w_i \text{ is one} \\ L_i & \text{if } S_i \text{ is even, } w_i \text{ is zero} \\ L_i + 1 & \text{if } S_i \text{ is even, } w_i \text{ is one and } L_i \neq -1 \\ L_i - 1 & \text{if } S_i \text{ is odd, } w_i \text{ is zero and } L_i = -1 \end{cases}$$

where w_i is the watermark bit of block$_i$ and L'_i is the marked LNZ level of selected block$_i$.

At the extraction phase in the decoder, the position of the LNZ level is needed to detect the selected blocks. Thus, the nonzero levels should not be changed to zero. Therefore, if the level value is equal to -1, it should be decremented instead on being incremented in case it needs to change.

To make our scheme robust against collusion attacks, instead of using a unique key in step 3, the key is generated based on MBs features [33]. These features need to be robust enough to avoid the possibility of variations after reencoding. To prevent computational complexity, we used the codec information for generating a key for each MB. In 4×4 intraprediction, nine modes are classified into three groups: 1) vertical and diagonal modes (0, 3, 4, 5, 7); 2) horizontal modes (1, 6, 8); and 3) dc mode (2). As similar modes may be changed to each other after reencoding, categorizing them makes the public key more robust in case of alternations. For three modes, two bits are needed and assigned for each mode. Thus, a 32-bit content-based key is generated for an MB, which includes 16 4×4 blocks. In addition, for 16×16 intraprediction, there are four modes for which, based on the prediction mode, a 32-bit content-based key is created. For example, in 4×4 intraprediction, for the first, second, and third groups, we can assign 00, 01, and 10, respectively. In addition, a 32-bit key that starts with 11 can be used for 16×16 intraprediction mode.

Some processes may change the intraprediction modes in blocks, which consequently lead to different level values, and therefore make the embedded watermark undetectable. After reencoding, the luma prediction modes may be changed, which affects the synchronization in the watermark extraction process. To evaluate this effect, we considered the rate of

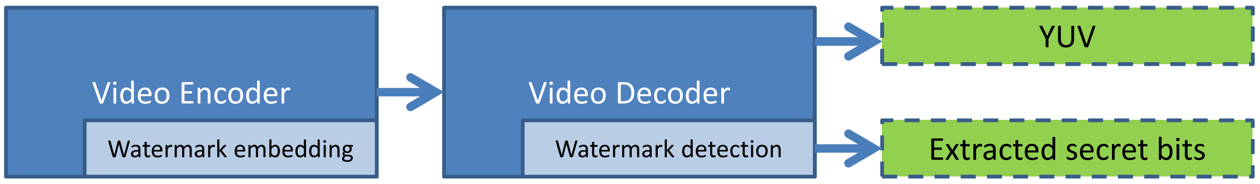

Fig. 1. Embedding and detecting flowchart.

prediction mode changes after reencoding. As it is shown in Table I, when the number of nonzero levels increases, the probability of changes in intramodes decreases. In other words, more textured blocks can withstand better against the reencoding process and, therefore, against other manipulations. These coefficients mostly correspond to nonflat areas, which are more vulnerable to attacks since attackers aim to change the texture areas, not the background.

As we classify intraprediction in groups and assign two bits to each group, the probability of group change is even less than changes in modes. Thus, the key can be extracted in most of the cases.

The embedding process does not alter the number of levels, which helps to keep the bitrate the same as the original without watermark embedding. Fig. 1 shows the embedding and detecting procedures. After decoding the stream, which includes watermark detection, the YUV data and the extracted secret bits are obtained.

We can easily adjust the properties of our watermarking system. By increasing k, the capacity is increased but transparency and robustness are decreased. Increasing k means using more blocks in each MB for embedding, which in turn raises the error rate in case of attacks or common processing operations. Therefore, increasing k decreases robustness of the system.

To remove redundancy, compression algorithms manipulate high-frequency elements more, since the human perceptual system is sensitive to changes in middle and low frequencies. To improve robustness, instead of embedding the watermark in only the LNZ levels (the highest frequency), other levels in the middle or even low frequencies can be used for embedding, although the distortion in this case would increase. When not only the LNZ level but also other nonzero levels are used for embedding, the capacity of the system is increased and it becomes more difficult for an attacker to track the changes.

In H.264/AVC compression, changing the quantization parameter (QP) results in varying the number of nonzero levels in MBs. It is evident that when QP is low, there are more nonzero levels compared with high QP. Therefore, if QP is low, we can embed 16 bits in each MB and if QP is high, we can embed less than 16 bits in each MB. In other words, if QP is high and compression rate is high (i.e., the video is very compressed), we need to choose a suitable k (less than 16). Thus, when QP is low, watermark embedding will result in less distortion and more capacity, and when QP is high, the provided capacity is lower.

D. Detecting

The embedded watermark bits are extracted in the video decoding process where the quantized DCT levels for each MB are entropy decoded. For each MB, the following steps result in extracting k embedded bits from each MB.

1) Sort the position values of the LNZ levels for 16 blocks of the current MB. Then, select k blocks that have higher nonzero position values and are used for embedding.
2) In each of the above selected blocks, a bit is embedded, which can be extracted as follows:

$$w'_i = \begin{cases} 0 & \text{if } S'_i \text{ is even} \\ 1 & \text{if } S'_i \text{ is odd} \end{cases}$$

where w'_i is the extracted bit of the ith selected block of the current MB in the decoder and S_i is the sum of all levels in the ith selected block of the current MB in the decoder.

To achieve the raw watermark stream for each MB, we need to use the encryption key of the current MB. This key was generated based on intraprediction modes in the encoder, which can be regenerated in the decoder as well.

In general, efficiency, complexity, energy usage, and simplicity are more important in the decoder implementation since it is at the client side with limited resources compared with the server side, which has more resources. In addition, some delay in the encoder is acceptable since it is done only once for a video. Thus, low complexity and simplicity in implementation are critical points in designing the decoder. One advantage of our proposed technique is its simplicity of implementation at the decoder side, allowing it to run for various real-time applications.

TABLE II
CONFIGURATION PARAMETERS OF THE JM SOFTWARE

Profile	Baseline
Level	3
Intra Period	1,3,5
Frame Rate	30
Rate Distortion Optimization	On
Number of encoded frames	100

IV. EXPERIMENTAL RESULTS AND ANALYSIS

Although our scheme can be used in any DCT-based video encoder, as a proof-of-concept, we have specifically implemented and integrated our scheme with the H.264/AVC reference software JM12.2 [36], although the presented results will be similar in other modern video codecs as well, and our conclusions are without loss of generality. Five standard video sequences from [37] (*Container, Foreman, Mobile, News,* and

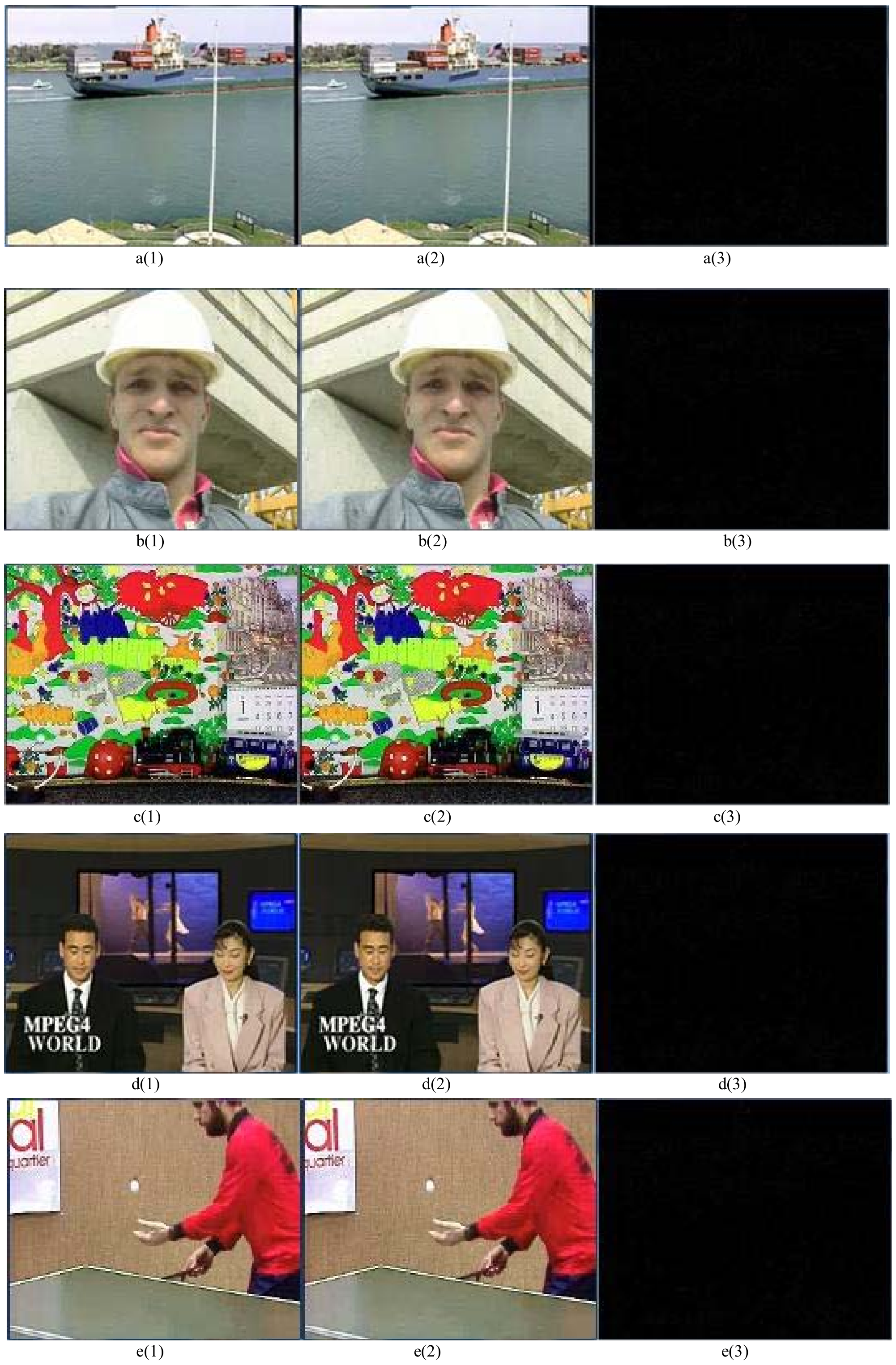

Fig. 2. Column 1: unmarked H.264/AVC compressed/decompressed frames (QP = 24 and 50th frame). Column 2: marked H.264/AVC compressed/decompressed frames. Column 3: the difference between the unmarked and marked decoded frame.

Tennis) in QCIF format (176 × 144 pixels) are used for our simulations. Since we embed the watermark in 16 × 16 MBs, our algorithm is independent of the resolution of the video, so there is no need to test higher resolution formats. Some important configuration parameters of our tests are given in Table II. The rest of the parameters have retained their default values.

The experimental results are divided into five parts. In the first part, the transparency and capacity of the proposed method are presented. The second and third parts show the robustness and security of the proposed scheme, respectively. Tampering results are presented in the fourth part, and finally some analysis and comparison with existing methods are provided in the fifth part.

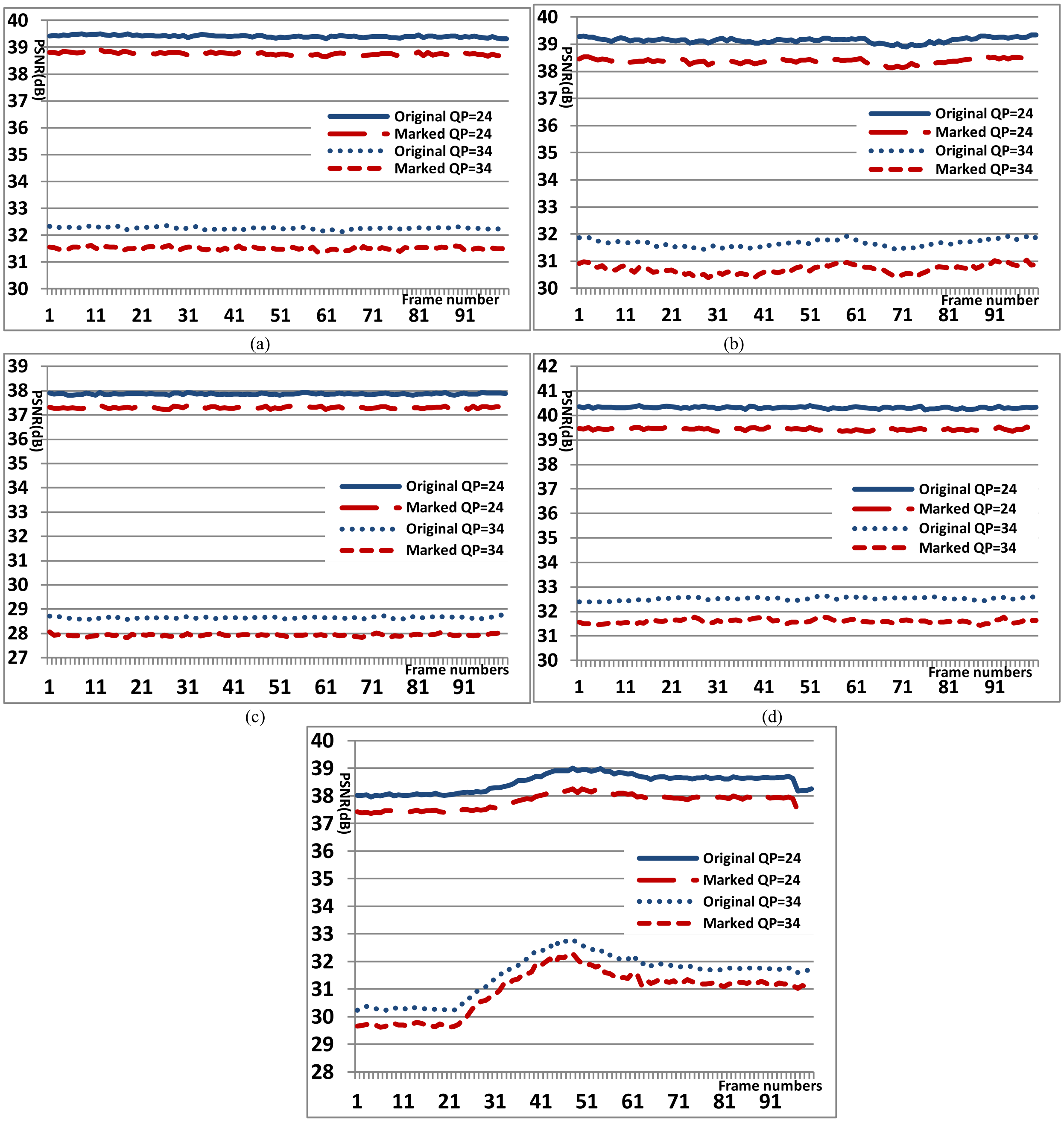

Fig. 3. Frame level PSNR before and after watermarking for (a) *Container*, (b) *Foreman*, (c) *Mobile*, (d) *News*, and (e) *Tennis*.

A. Transparency and Capacity

To measure the transparency of the proposed system, subjective and objective techniques can be used. Fig. 2a(1)–e(1) shows unmarked H.264 compressed/decompressed frames of the test sequences, while Fig. 2a(2)–e(2) shows watermarked H.264 compressed/decompressed frames, and Fig. 2a(3)–e(3) shows the difference between the two when k is eight. It is obvious from these figures that no significant visible distortion can be observed in any sequences, meeting the transparency requirement explained in Section III.

In addition to these subjective tests, objective measurements help us to prove the transparency of the embedding process. Fig. 3 shows the luma PSNR variation for 100 frames and the effect of embedding the secret information in the frames. The blue line in the plots shows the PSNR after H.264/AVC compression without watermarking, whereas the red line shows the PSNR after H.264/AVC compression and watermarking. In other words, we used the unmarked and compressed video sequences as the original video clips and the watermarked and compressed video sequences as marked. Please note that the results for two QP values (i.e., 24 and 34) are presented in this figure.

While PSNR is a very popular and widely used evaluation method, it is known that its correlation to subjective quality measures is not always good, because it reflects only the luminance component and neglects the chrominance component,

TABLE III
TRANSPARENCY AND BITRATE FOR AVERAGE OF 100 FRAMES FOR FIVE VIDEO SEQUENCES UNDER THE SAME QP WITH INTRAPERIOD = 1

		QP=10		QP=24		QP=34	
		Original	Marked	Original	Marked	Original	Marked
Container	PSNR (dB)	51.09	50.62	39.41	38.75	32.26	31.5
	SSIM	0.9960	0.9956	0.9577	0.9518	0.8978	0.8896
	Bitrate (kbps)	3460.2	3482.7	1262.7	1289.9	532.6	554.3
Foreman	PSNR (dB)	51.04	50.59	39.15	38.38	31.66	30.71
	SSIM	0.9971	0.9967	0.9692	0.9627	0.9041	0.8870
	Bitrate (kbps)	3786.6	3807.6	1419.4	1455.2	599.7	634.4
Mobile	PSNR (dB)	51	50.55	37.87	37.3	28.65	27.93
	SSIM	0.9993	0.9992	0.9893	0.9876	0.9280	0.9174
	Bitrate (kbps)	6469.1	6474.6	3241.4	3259.4	1629.8	1673.3
News	PSNR (dB)	51.31	50.77	40.32	39.44	32.52	31.6
	SSIM	0.9971	0.9967	0.9811	0.9764	0.92917	0.91388
	Bitrate (kbps)	3235.2	3264.6	1306.3	1339.1	599.69	625.63
Tennis	PSNR (dB)	51	50.56	38.49	37.8	31.51	30.95
	SSIM	0.9967	0.9963	0.9482	0.9408	0.7951	0.78553
	Bitrate (kbps)	3919.8	3934.3	1448.8	1489.1	493.7	512.1
Average	PSNR (dB)	51.01	50.62	39.05	38.33	31.32	30.54
	SSIM	0.9972	0.9969	0.9691	0.9639	0.8908	0.8787
	Bitrate (kbps)	4174.1	4192.7	1735.7	1766.54	771.1	799.9

which is important to human perception. Therefore, in addition to PSNR, we have also used the structural similarity index (SSIM) [38], which is a more advanced measurement index. While PSNR measures errors between the original and marked image, SSIM measures the structural distortion, luminance, and contrast differences between the two frames. The idea behind SSIM is that the human vision system is specialized in extracting structural information from the viewing field and it is not specialized in extracting the errors. Therefore, a measurement on structural distortion can give a better correlation to subjective impressions. SSIM is in the range zero to one, where zero shows zero correlation, i.e., the reference frame is entirely different from the target, and one shows that they are identical. Table III shows the PSNR, SSIM, and bitrate results of our method for 100 frames of each test sequence with three different QP values. The last three rows are the average of the five sequences for PSNR, SSIM, and bitrate, respectively. The overall average results for all three QPs show a quality degradation of 0.65 dB in term of PSNR, and 0.0058 for SSIM. In addition, the bitrate is increased by 0.019, on average.

Table IV shows transparency and bitrate similar to Table III, however, with different intraperiods. For intraperiod equal to five, the average quality loss is 0.29 dB in term of PSNR and 0.003 for SSIM, while bitrate is increased by 0.023. It is evident that, increasing the intraperiod leads to better transparency and lower computation time; however, it decreases the efficiency of the tamper detection system since the distance between I-frames is increased. However, for general applications, it is satisfactory. If all frames are I-frames, we get the worst transparency. However, if P frames are used too, which is the case in reality, the distortion of the marked video sequence will decrease and the transparency of our scheme is even more, because P frames are not used for watermark embedding.

By embedding and modifying frames, both transparency and bitrate are changed. To show the effect of embedding on just transparency, the bitrate obtained by compression with a fixed QP of the original sequence is used to compress the watermarked sequence, i.e., we compress the original sequence with a fixed QP and get a certain bitrate. Then, in the watermarking process, we compress the sequence with that bitrate, which means that the bitrates in the original and marked sequences are practically the same and just transparency is different. Table V shows the quality in terms of PSNR and SSIM and bitrate for *Container* and the average of the five test sequences. As the average rows illustrate, the average PSNR degradation is 0.88 dB, while it is 0.0090 for SSIM. Furthermore, the bitrate increase is just 0.05%, whereas we configured the coder to have the same bitrate. To illustrate alterations between the original and the marked bitrates, the differences are provided in normal and percentage values.

As PSNR and SSIM do not consider the temporal activity, we have also measured the visual quality metric (VQM) [39], which provides an objective measurement for the perceived video quality. VQM measures the perceptual effects of video impairments including global noise, blurring, jerky/unnatural motion, block distortion, and color distortion, and combines

TABLE IV
TRANSPARENCY AND BITRATE FOR AVERAGE OF 100 FRAMES FOR FIVE VIDEO SEQUENCES WITH DIFFERENT INTRAPERIODS

		QP=10		QP=24		QP=34	
		Original	Marked	Original	Marked	Original	Marked
Container (IPP)	PSNR (dB)	50.08	49.94	39.02	38.54	32.18	31.5
	SSIM	0.99513	0.99497	0.95531	0.9507	0.89714	0.89025
	Bitrate	1876.07	1897.34	452.7	470.84	166.22	179.07
Container (IPPPP)	PSNR (dB)	49.9	49.8	38.7	38.4	32.0	31.5
	SSIM	0.994904	0.994833	0.9528473	0.9491725	0.8961765	0.8895265
	Bitrate	1586.62	1601.22	317.29	329.82	105.68	113.92
Foreman (IPP)	PSNR (dB)	49.98132	49.85061	38.63386	38.2061	32.17789	31.40606
	SSIM	0.9964462	0.9963205	0.9668227	0.9623607	0.905182	0.8919295
	Bitrate	2322.7	2336.75	514.51	540.53	169.86	184.57
Foreman (IPPPP)	PSNR (dB)	49.78	49.72	38.43	38.13	32.06	31.45
	SSIM	0.9963	0.99624	0.96584	0.96259	0.9043	0.8935
	Bitrate	2103.08	2111.05	399.85	415.9	118.3	128.34
Mobile (IPP)	PSNR (dB)	49.72152	49.59698	36.56911	36.4066	27.61132	27.33183
	SSIM	0.9991335	0.9991054	0.986376	0.985697	0.9114812	0.9058768
	Bitrate	4975.54	4979.69	2115.45	2126.99	768.89	798.32
Mobile (IPPPP)	PSNR (dB)	49.48573	49.41936	36.32265	36.23286	27.39994	27.21657
	SSIM	0.9990901	0.9990738	0.9857422	0.9853247	0.9043035	0.8935036
	Bitrate	4713.17	4716.41	1905.3	1913.66	118.3	128.34
News (IPP)	PSNR (dB)	50.65795	50.37802	40.07405	39.41684	32.63514	31.83452
	SSIM	0.9968371	0.9965508	0.9805793	0.9766553	0.9294591	0.9168145
	Bitrate	1433.46	1458.86	445.95	463.54	191.05	205.11
News (IPPPP)	PSNR (dB)	50.51308	50.29208	39.91827	39.35188	32.55128	31.85107
	SSIM	0.9967507	0.9965182	0.9802962	0.9767487	0.9286135	0.9177506
	Bitrate	1110.48	1126	305.48	317.14	123.63	132.18
Tennis (IPP)	PSNR (dB)	49.87072	49.7545	37.6214	37.26951	31.10055	30.71341
	SSIM	0.9958363	0.995717	0.9382944	0.9337276	0.7847419	0.7776902
	Bitrate	2780.08	2791.14	732.61	758.1	198.67	210.21
Tennis (IPPPP)	PSNR (dB)	49.6565	49.59025	37.32362	37.11013	30.843	30.55712
	SSIM	0.9956349	0.9955639	0.9339122	0.9313001	0.7778066	0.7717371
	Bitrate	2609.54	2617.97	636.4	653.57	161.95	168.59
Average (IPP)	PSNR (dB)	50.062	49.904	38.382	37.97	31.142	30.556
	SSIM	0.996678	0.996534	0.965476	0.96183	0.8856	0.876512
	Bitrate	2677.57	2692.756	852.244	872	298.938	315.456
Average (IPPPP)	PSNR (dB)	49.862	49.766	38.15	37.842	30.978	30.504
	SSIM	0.996534	0.996444	0.963728	0.961026	0.882914	0.875242
	Bitrate	2424.578	2434.53	712.864	726.018	222.3	232.94

them into a single metric. VQM has a high correlation with subjective video quality assessment, and is between zero and one where zero shows not having any distortions and one presents maximum impairment. We compare the unmarked and compressed video sequences with the watermarked and compressed video sequences. Table VI shows the effect of our watermarking system on the videos. For example, embedding in *Container* with QP = 24 results in 0.48-dB distortion in video quality and 0.04% increase in bitrate.

It should be noted again that the properties (transparency, robustness, capacity, and security) of the proposed watermarking system are adjustable. As explained in Section III-B, in the embedding part, each MB has potential to embed 16 bits inside it, i.e., 16 bits in each MB and 99 $\times$ 16 bits in each

TABLE V
TRANSPARENCY AND BITRATE FOR AVERAGE OF 100 FRAMES FOR FIVE VIDEO SEQUENCES UNDER THE SAME BITRATE

		Original QP=10	Marked	difference	Original QP=24	Marked	difference	Original QP=34	Marked	difference
Container	PSNR (dB)	50.24	49.51	0.73	38.94	38.17	0.77	31.09	29.95	1.14
	SSIM	0.9938	0.9917	0.00206	0.9509	0.9444	0.0064	0.8692	0.8533	0.0159
	Bitrate kbps	3313.2	3328.5	15.3	1236.3	1239.5	3.2	534.1	532.1	–2
				0.46 %			0.25%			– 0.39%
Foreman	PSNR (dB)	50.46	49.75	0.71	38.87	37.83	1.04	30.64	29.42	1.22
	SSIM	0.99679	0.99619	0.0006	0.96384	0.95467	0.00916	0.84719	0.82100	0.02620
	Bitrate	3786.6	3785.62	– 0.98	1419.71	1419.61	0.1	600.02	599.99	0.03
	kbps			–0.02%			– 0.007%			–0.005%
Mobile	PSNR (dB)	50.77	50.27	0.5	37.51	36.86	0.65	28.2	27.16	1.04
	SSIM	0.99926	0.99914	0.00011	0.98636	0.98390	0.00246	0.89378	0.87443	0.01935
	Bitrate kbps	6457.43	6467.51	10.08	3233.56	3236.48	2.92	1628.73	1627.82	-0.93
				0.15%			0.09%			-0.06%
News	PSNR (dB)	50.81	50.09	0.72	39.82	38.66	1.16	31.4	30.18	1.22
	SSIM	0.99665	0.99600	0.00065	0.97443	0.96689	0.00753	0.88518	0.86415	0.02103
	Bitrate kbps	3171.19	3166.32	–4.87	1300.31	1299.16	-1.15	599.63	599.81	0.18
				-0.15%			-0.089%			0.03%
Tennis	PSNR (dB)	50.75	50.02	0.73	38.25	37.27	0.98	30.31	29.56	0.75
	SSIM	0.99610	0.99546	0.00064	0.93070	0.91736	0.01334	0.75398	0.74284	0.01115
	Bitrate kbps	3918.03	3914.92	–3.11	1446.64	1447.41	0.77	493.86	494	0.14
				-0.08%			0.05%			0.03%
Average	PSNR (dB)	50.60	49.92	**0.67**	38.67	37.75	**0.92**	30.32	29.25	**1.07**
	SSIM	0.9965	0.9957	**0.0008**	0.9612	0.9534	**0.0077**	0.8498	0.8311	**0.0187**
	Bitrate kbps	4129.3	4132.5	**3.2**	1727.3	1728.4	**1.1**	771.3	770.7	**–0.6**
				0.0715%			**0.06%**			**-0.08%**

frame. In the worst case, if an MB does not have enough suitable blocks for embedding, the most significant bit of the index (in our case MBs number) is embedded in available vacancies. The parameters of our watermarking system can be chosen based on application requirements. In our experimental results, to embed eight bits in each MB, eight blocks of each MB is selected. Seven bits of these eight bits are sufficient for the index of each MB since there are 99 MBs in a frame. In addition, one bit of each MB is used for the index of the current frame and thus there is 99 bits that can be used for the index of the frame.

B. Robustness

In this section, the robustness of our method against common signal processing operations (recompression, additive white Gaussian noise, and brightness increase) is analyzed. The robustness is measured by bit error rate (BER) between the extracted watermark and the original watermark. Fig. 4 shows the encoder along with its embedding function at the left-hand side, while the decoder along with its detection function is located at the right-hand side. To study the effects of common signal processing operations, an attack block is added, shown with a red block in Fig. 4. It is evident that to change the marked sequence, the sequence needs to be decoded from H.264/AVC to raw video, manipulated, and then encoded back to H.264/AVC by the original coder.

Table VII shows the BER after recompression with various QPs, Gaussian noise, salt and pepper noise, and increasing brightness. White noise of mean zero and different variances are added to each frame of the video sequence. In addition, salt and pepper noise with different noise densities are studied. Finally, increasing brightness is considered. The table shows the effects of these operations on the BER between the extracted and the original watermark. As can be seen, the proposed method can resist common signal processing attacks, although error correction codes can further improve them effectively.

We test the robustness of the watermark against transcoding procedures to verify whether the embedded watermark can survive when the quantization level is changed or a lower bitrate is employed during the transcoding. We can see that the watermark detections will be affected using different QP values since the LNZ positions are changed. However,

TABLE VI
EFFECTS OF WATERMARKING EMBEDDING ON VIDEO CLIPS

		QP=10	QP=24	QP=34
Container	PSNR (dB)	0.14	0.48	0.68
	SSIM	0.00016	0.0046	0.0068
	VQM	0.0093	0.0031	0.0249
	Bitrate (kbps)	0.0113	0.0400	0.0773
Foreman	PSNR (dB)	0.13	0.42	0.77
	SSIM	0.00013	0.0044	0.0132
	VQM	0.0003	0.0039	0.0017
	Bitrate (kbps)	0.0060	0.0505	0.0866
Mobile	PSNR (dB)	0.12	0.16	0.28
	SSIM	0.0001	0.0006	0.0056
	VQM	0.0002	0.0017	0.0034
	Bitrate (kbps)	0.0008	0.0054	0.0382
News	PSNR (dB)	0.28	0.65	0.81
	SSIM	0.0002	0.0039	0.0126
	VQM	0.0011	0.0008	0.0069
	Bitrate (kbps)	0.0177	0.0394	0.0735
Tennis	PSNR (dB)	0.12	0.35	0.39
	SSIM	0.0001	0.0045	0.0070
	VQM	0.0001	0.0049	0.0127
	Bitrate (kbps)	0.0039	0.0347	0.0580

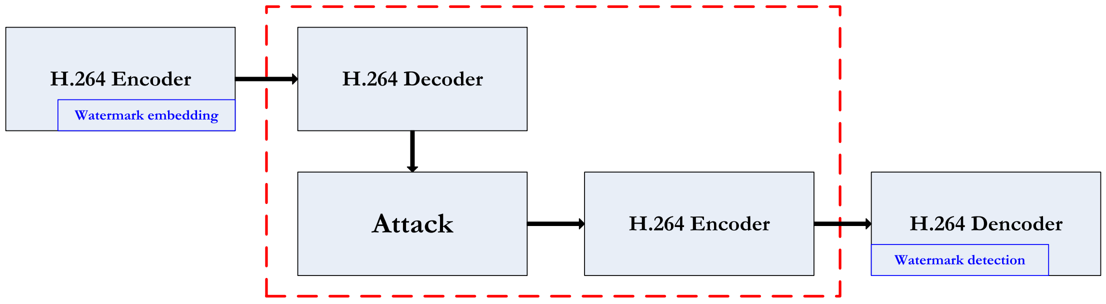

Fig. 4. Flowchart of embedding, attack, and detecting.

it should be noted that the video quality becomes poor when the QP is changed. To provide a secure method to detect tampering, in our system, the encoder and the recompression modules use the same QP values; i.e., QP in the encoder and decoder is the same.

C. Security

To provide a secure method, pseudorandom number generators are used to change the secret bit stream to another stream, which makes it more difficult for an attacker to extract the secret information. The watermark bit stream is constructed as the XOR of the raw watermark and a key [40]. Fig. 5 shows the encryption and decryption flowchart. To make it even more secure, the key is generated based on the texture of each MB, thus each MB has its own key. Using different QP in the encoder and the decoder results in changing intraprediction modes in the decoding, thus the key will be different and the watermark stream cannot be extracted correctly [33]. When the number of nonzero levels is high, the probability of changes in intramodes is less than 5%. Therefore, texture blocks that are important in authentication can resist better against reencoding and other manipulations.

With this security scheme, an attacker has following difficult challenges to avoid tamper detection.

1) The attacker needs to have both the original and marked video to find the modified QDCT values. This is highly unlikely that an attacker can access the original video to discover the changes.
2) In the very rare case that an attacker has access to the original video clip and finds the modified QDCT, the attacker must keep the LNZ level and change the other

TABLE VII
BER (IN %) BETWEEN THE EXTRACTED AND THE ORIGINAL WATERMARK, AFTER RECOMPRESSION

sequence	QP	H.264/AVC recompression	Gaussian noise		Salt & pepper noise		Brightness increase	
			σ = 0.1	σ = 0.0 1	0.1	0.0 1	1	2
Container	24	81.0	–	–	64.1	74.2	63.8	56.6
	34	82.2	–	65.1	66	75.1	73.1	64
	44	88.0	71.1	75.3	75.1	77.8	76.1	75.8
Foreman	24	77.9	–	–	60.5	73.2	58.7	51.5
	34	79.2	–	61.2	60.1	69.2	69.2	61.3
	44	85.7	50.6	53.6	52.1	53.9	54.1	53.5
Mobile	24	71.4	–	–	60.8	71.6	57.1	47.3
	34	76.3	–	65.8	69.1	78.5	77.3	67.8
	44	83.0	74.1	78.6	76.1	78.2	78.9	78.3
News	24	68.0	–	–	55.7	67.9	53.7	50.1
	34	72.0	–	63.1	63.2	70.5	70.1	68.3
	44	86.1	77.2	81.9	80.1	82.5	83.9	83.2
Tennis	24	83.0	–	–	70.0	79.8	75.1	67.1
	34	85.0	–	70.9	66.2	75.1	74.2	72.1
	44	87.9	61.5	64	63.2	65.8	64.7	63.8
Average	34	80.4	66.9	68	65.5	72.9	68.7	64.1

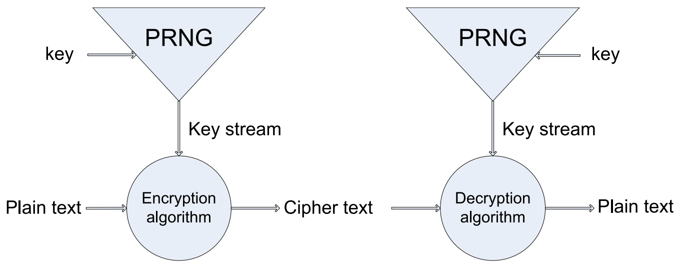

Fig. 5. Encryption and decryption algorithms.

levels to prevent tamper detection. However, this is very difficult, if not practically impossible, because:

a) the embedding and extracting processes work based on the sum of all levels in each selected block;
b) the encryption key is generated based on the levels and content of the blocks. Thus, any change in the levels will be detected at our tamper detection phase.

Therefore, it is practically impossible for an attacker to bypass the tampering detection stage, which is described next.

D. Tampering Detection

Tampering can be classified into spatial, temporal, or both. Spatial tampering refers to modifications in the frame, such as cropping and replacement, content adding and removal. In authentication, we must determine which parts of the frame have been modified and to identify the reasons for those modifications, i.e., changes in a frame helps us to detect if the reason is a common video processing operation (such as compression) or an attack.

Table VIII shows the bit correct rate (BCR) of the proposed scheme against recompression for different QPs. As the last row illustrates, the average BCR is 0.81, which is quite good. This table shows the accuracy of our method when there is no attack in the red block of Fig. 4. However, some sparse binary errors caused by transmission, storage, or data processing can happen and distort the video. These should not be mistaken as tampering. Fig. 6 shows the effect of recompression ($QP = 24$) on 12 frames of the *Tennis* video sequence. As it is evident, those MBs whose index could not be extracted (the black blocks) are spread randomly in the frames. These frames and black MBs show that the error happened by a general signal processing process (here recompression) and there is no malicious attack.

On the other hand, as Fig. 7 shows, if the MBs, which contain the critical area, such as the ball area, were changed in a significant number of frames, it is a sign that an attacker manipulated some frames. Thus, based on the analyses, it is possible to judge the type of modification. We can see that, without having the original video and only by analyzing the received video, if errors are spread in different areas of frames randomly, it is evident that there was no malicious attack and it was the result of noise, recompression, filtering, and so on. However, if errors are concentrated in a specific area, an attack has occurred.

To detect an attack, we need to consider the error, which continuously occurs in the same MBs in a series of frames. Random errors that are not repeated in frames are skipped. For example, the MB in the second row and fifth column of a(7) in Fig. 7 has repeated in the first seven frames. Thus, it has not randomly happened and is an attack.

Without any temporal manipulation, the extracted frame number is the same as the observed frame. We have tested frame dropping by removing frames 61–80 of the *Tennis* video

TABLE VIII
BCR FOR DIFFERENT QP

Sequence	coded QP / Recompressed QP	BCR (%)
Container	24/24	81.6
	34/34	82.3
	44/44	88.2
Foreman	24/24	78.3
	34/34	79.4
	44/44	85.7
Mobile	24/24	73.9
	34/34	77.3
	44/44	83.1
News	24/24	68.5
	34/34	72.1
	44/44	86.2
Tennis	24/24	83.5
	34/34	85.1
	44/44	87.9
Average	–	**80.9**

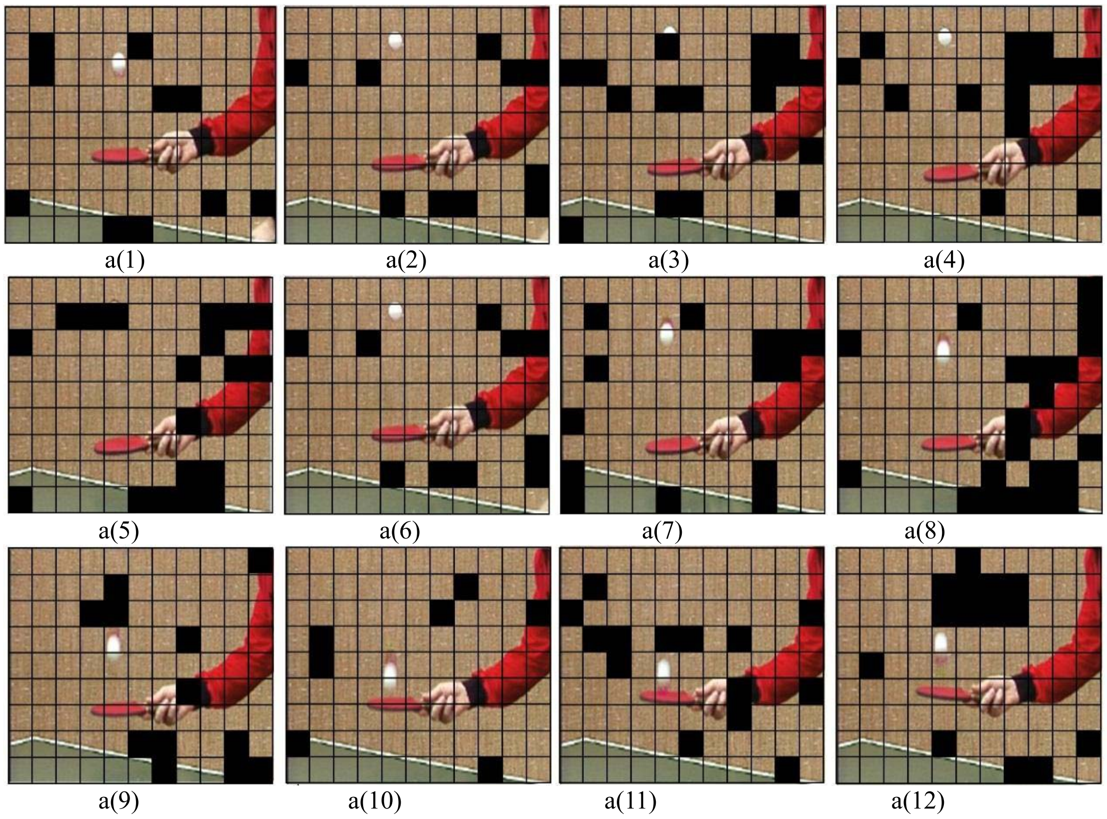

Fig. 6. Tampering detection of the first 12 frames of *Tennis*.

sequence. Fig. 8(a) shows a jump, which proves that 20 frames, starting from the 61st frame, are missing. In addition to frame dropping, we also tested frame replacing by replacing frames 21–40 by frames 1–20, and frames 40–50 by frames 50–60. Fig. 8(b) shows that our method can detect such changes successfully.

In summary, our experimental results show the following properties of our tampering detection method.

1) When the video is encoded and decoded and then encoded and decoded with watermark extraction, the performance of the system is 80.9%, as shown in Table VIII.
2) When the decoded stream is not just played but is also modified, the performance is decreased based on the type of modification, as shown in Table VII.

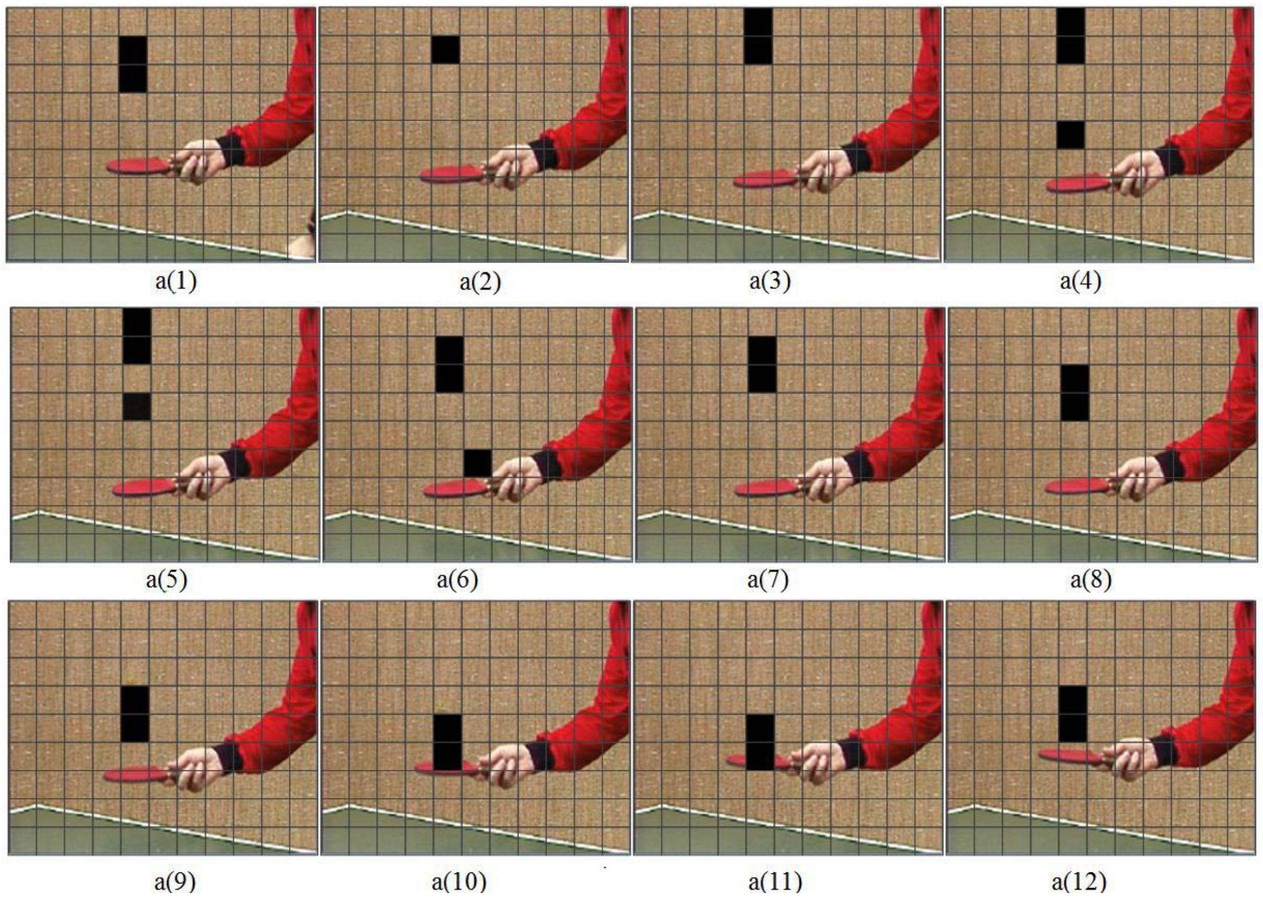

Fig. 7. Tampering detection of the first 12 frames of *Tennis* after a malicious attack.

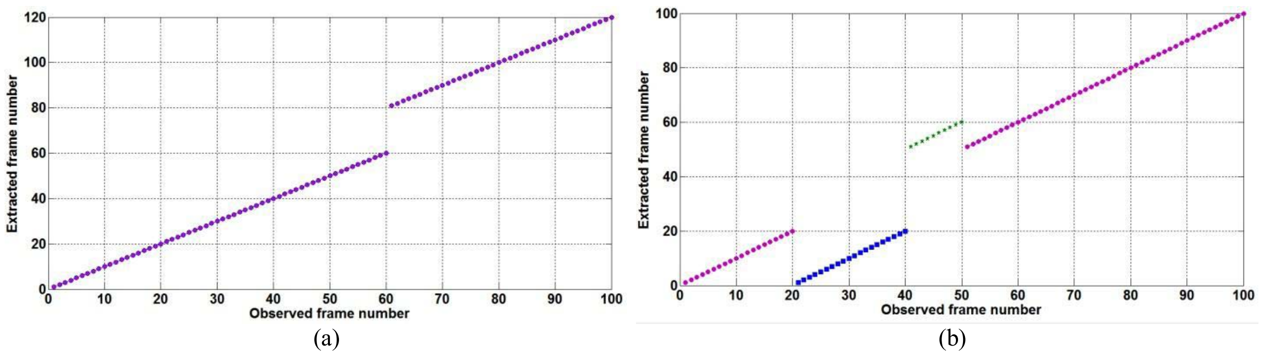

Fig. 8. Temporal tampering (a) frame dropping and (b) frame replacing.

3) What Figs. 6 and 7 show is that the meaning of the data is important. In other words, sparse binary errors are affected by some processing that do not change the meaning of the media. However, changes in specific parts of frames show malicious attacks.

E. Further Analysis and Comparison With the Existing Methods

The performance of any watermarking system can be improved by applying ECCs. In our system, one bit in each MB is assigned for frame indexing, as explained before. We need 10 bits for frame indexing, and there are about 99 bits to embed the frame's index. For example, using ECC, 10 bits can be converted to 15 and the 15 bits repeated five times in each frame. The type of ECC and repeating depends on the application's requirements. For instance, surviving against Gaussian noise needs a good ECC whereas increasing the repeating time increases robustness against burst errors. In our experimental results, to get robustness against H.264/AVC recompression, it was sufficient to use the repeating idea. After H.264/AVC recompression, the frames' indices are extracted successfully.

While it is difficult to compare in a fair manner our scheme with others, since each scheme has its own special properties under different conditions, we nevertheless compare some existing methods with our proposed scheme as follows.

Similar to our scheme, the work in [5] is also in the H.264/AVC domain but it has the following properties.

1) Reference [5] is slower than our approach since its watermark embedding method is more complex than ours. Our proposed method simply changes the nonzero levels based on the sum of all levels in the current block; however, in [5], before performing the embedding process, features of the video frames should be derived. This makes [5] more complicated and increases the computation time thus decreasing the speed.
2) At the same distortion level, the bitrate in [5] increases about 7.13% compared with 0.06% in our method. For example, if the original bitrate is 800 kb/s, using [5], we will need to support a 56.8-kb/s increase of the bitrate, compared with only 0.48-kb/s increase using our method, which is orders of magnitude better than [5].
3) Reference [5] provides better robustness against recompression, Gaussian noise, and brightness increase; however, its robustness against salt and pepper noise has not be reported. Our proposed scheme and [5] both work perfectly against temporal attacks, such as frame dropping.
4) The experimental results in [5] are provided for only $QP = 28$, whereas we simulate the fidelity of our proposed system for a wide range of QPs.

For robustness, we note that methods such as [22]–[24] are fragile and not robust against H.264/AVC recompression, as their watermark is lost after recompression. Other methods such as [1] are in the uncompressed domain and cannot be applied directly to H.264/AVC. Therefore, objective comparison between our method and existing ones is unfair and not meaningful. Furthermore, in [1], the payloads of the robust watermark and the fragile watermark are 32 and 1024 bits, respectively.

In addition to the above, our experience leads to the following points.

1) To achieve more robustness, instead of using the LNZ coefficient, other nonzero levels can be used, as mentioned in Section III.
2) Using nonzero QDCT levels is more robust compared with schemes that use DCT before quantization, since some of the modified levels may convert to zero after quantization.
3) In our proposed scheme, nonzero levels are modified. Changing a zero level to a nonzero level can change the bitrate enormously since context-adaptive variable-length coding is very sensitive to the levels.
4) Our scheme is robust against attacks, such as dropping, jittering, and delay, since extracting and detecting the secret bits are only based on each single frame and independent from other frames. This is very useful for networked applications where these attacks can happen frequently.

V. Conclusion

A practical system of digital video watermarking is suggested for authenticating and tampering detection of compressed videos. To design an efficient and low-complexity method, the embedding and extracting of watermarks are integrated with the coding and decoding routines of the video codec. To assure transparency to the human visual system, the MBs' and frames' indices are embedded into the LNZ quantized DCT value of the blocks. The suggested authentication method provides detection of spatial, temporal, and spatiotemporal tampering. The experimental results show that the distortion caused by our system is very low on average, PSNR is −0.88 dB, SSIM is −0.0090, increasing bitrate is just 0.05%, and BCR after H.264/AVC recompression is 0.71–0.88. Adding content-based cryptography to the watermarking system increases the security of the system and slightly decreases BCR (1% to 5%) after H.264/AVC recompression. Furthermore, to distinguish malicious attacks from common video processing operations, such as H.264/AVC recompression, noise, and brightness increasing, analysis of the error is used to detect tampering.

References

[1] S. Chen and H. Leung, "Chaotic watermarking for video authentication in surveillance applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 5, pp. 704–709, May 2008.

[2] B. Zhu, M. Swanson, and A. Tewfik, "When seeing isn't believing [multimedia authentication technologies]," *IEEE Signal Process. Mag.*, vol. 21, no. 2, pp. 40–49, Mar. 2004.

[3] F. Bartolini, A. Tefas, M. Barni, and I. Pitas, "Image authentication techniques for surveillance applications," *Proc. IEEE*, vol. 89, no. 10, pp. 1403–1418, Oct. 2001.

[4] P.-C. Su, C.-S. Wu, I.-F. Chen, C.-Y. Wu, and Y.-C. Wu, "A practical design of digital video watermarking in H.264/AVC for content authentication," *Signal Process, Image Commun.*, vol. 26, nos. 8–9, pp. 413–426, Oct. 2011.

[5] D. Xu, R. Wang, and J. Wang, "A novel watermarking scheme for H.264/AVC video authentication," *Signal Process., Image Commun.*, vol. 26, no. 6, pp. 267–279, 2011.

[6] J. Fridrich, M. Goljan, and A. C. Baldoza, "New fragile authentication watermark for images," in *Proc. ICIP*, vol. 1. Vancouver, BC, Canada, 2000, pp. 446–449.

[7] K. Maeno, Q. Sun, S.-F. Chang, and M. Suto, "New semi-fragile image authentication watermarking techniques using random bias and nonuniform quantization," *IEEE Trans. Multimedia*, vol. 8, no. 1, pp. 32–45, Feb. 2006.

[8] C. Fei, D. Kundur, and R. H. Kwong, "Analysis and design of secure watermark-based authentication systems," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 1, pp. 43–55, Mar. 2006.

[9] J. Sang and M. S. Alam, "Fragility and robustness of binary phase-only filter based fragile/semi-fragile digital image watermarking," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 3, pp. 595–606, Mar. 2008.

[10] M. Fallahpour, M. Semsarzadeh, S. Shirmohammadi, and J. Zhao, "A realtime spatio-temporal watermarking scheme for H.264/AVC," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf.*, Minneapolis, MN, USA, May 2013, pp. 872–875.

[11] K. S. Wong, K. Tanaka, K. Takagi, and Y. Nakajima, "Complete video quality-preserving data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 10, pp. 1499–1512, Oct. 2009.

[12] R. Iqbal, S. Shirmohammadi, A. E. Saddik, and J. Zhao, "Compressed-domain video processing for adaptation, encryption, and authentication," *IEEE Multimedia*, vol. 15, no. 2, pp. 38–50, Apr./Jun. 2008.

[13] J. Zhao, W. J. Tam, S. Wang, D. Zheng, and F. Speranza, "A digital watermarking and perceptual model based video quality measurement," in *Proc. IEEE Conf. Instrum. Meas. Technol.*, May 2005, pp. 1729–1734.

[14] M. Barni, F. Bartolini, and N. Checcacci, "Watermarking of MPEG-4 video objects," *IEEE Trans. Multimedia*, vol. 7, no. 1, pp. 23–32, Feb. 2005.

[15] S. Biswas, S. R. Das, and E. M. Petriu, "An adaptive compressed MPEG-2 video watermarking scheme," *IEEE Trans. Instrum. Meas.*, vol. 54, no. 5, pp. 1853–1861, Oct. 2005.

[16] S. N. Biswas, S. Nahar, S. R. Das, E. M. Petriu, M. H. Assaf, and V. Groza, "MPEG-2 digital video watermarking technique," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf.*, May 2012, pp. 225–229.

[17] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[18] B. G. Mobasseri and M. J. Sieffert, "Content authentication and tamper detection in digital video," in *Proc. IEEE Int. Conf. Image Process.*, Vancouver, BC, Canada, 2000, pp. 458–461.

[19] Q. B. Sun, D. J. He, Z. S. Zhang, and Q. Tian, "A secure and robust approach to scalable video authentication," in *Proc. Int. Conf. Multimedia Expo*, vol. 2. Jul. 2003, pp. 209–212.

[20] X. L. Chen and H. M. Zhao, "A novel video content authentication algorithm combined semi-fragile watermarking with compressive sensing," in *Proc. 2nd Int. Conf. Intell. Syst. Des. Eng. Appl.*, Sanya, Hainan, China, Jan. 2012, pp. 134–137.

[21] Y. Shi, M. Qi, Y. Yi, M. Zhang, and J. Kong, "Object based dual watermarking for video authentication," *Int. J. Light Electron Opt.*, vol. 124, no. 19, pp. 3827–3834, 2013.

[22] D. K. Zou and J. A. Bloom, "H.264/AVC stream replacement technique for video watermarking," in *Proc. IEEE ICASSP*, Las Vegas, NV, USA, Apr. 2008, pp. 1749–1752.

[23] S. K. Kapotas and A. N. Skodras, "Real time data hiding by exploiting the IPCM macroblocks in H.264/AVC streams," *J. Real-Time Image Process.*, vol. 4, no. 1, pp. 33–41, Mar. 2009.

[24] T. Y. Kuo and Y. C. Lo, "Fragile video watermarking technique by motion field embedding with rate-distortion minimization," *J. Commun. Comput.*, vol. 6, no. 1, pp. 16–23, 2009.

[25] C. H. Fei, D. Kundur, and R. H. Kwong, "Analysis and design of secure watermark-based authentication systems," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 1, pp. 43–55, Mar. 2006.

[26] Q. B. Sun, D. J. He, and Q. Tian, "A secure and robust authentication scheme for video transcoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 10, pp. 1232–1244, Oct. 2006.

[27] J. Dittmann, A. Steinmetz, and R. Steinmetz, "Content-based digital signature for motion pictures authentication and content fragile watermarking," in *Proc. IEEE Int. Conf. Multimedia Comput. Syst.*, vol. 2. 1999, pp. 209–213.

[28] C.-C. Lai and C.-C. Tsai, "Digital image watermarking using discrete wavelet transform and singular value decomposition," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 11, pp. 3060–3063, Sep. 2010.

[29] X. Ma, Z. Li, H. Tu, and B. Zhang, "A data hiding algorithm for H.264/AVC video streams without intra-frame distortion drift," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 10, pp. 1320–1330, Oct. 2010.

[30] H.-Y. Huang, C.-H. Yang, and W.-H. Hsu, "A video watermarking technique based on pseudo-3-D DCT and quantization index modulation," *IEEE Trans Inf. Forensics Security*, vol. 5, no. 4, pp. 625–637, Dec. 2010.

[31] M. A. Suhail and M. S. Obaidat "Digital watermarking-based DCT and JPEG model," *IEEE Trans. Instrum. Meas.*, vol. 52, no. 5, pp. 1640–1647, Oct. 2003.

[32] J. Zhang, A. T. S. Ho, G. Qiu, and P. Marziliano, "Roubust video watermarking of H.264/AVC," *IEEE Trans. Circuits Syst.*, vol. 54, no. 2, pp. 205–209, Feb. 2007.

[33] A. Mansouri, A. M. Aznaveh, F. Torkamani-Azar, and F. Kurugollu, "A low complexity video watermarking in H.264 compressed domain," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 649–657, Dec. 2010.

[34] X. Gong and H. Lu, "Towards fast and robust watermarking scheme for H.264 video," in *Proc. IEEE Int. Symp. Multimedia*, Dec. 2008, pp. 649–653.

[35] D. Wang, S. Huang, G. Feng, and S. Wang, "Perceptual differential energy watermarking for H.264/AVC," *Multimedia Tools Appl.*, vol. 60, no. 3, pp. 537–550, 2012.

[36] (2013, Dec.). *JVT Reference Software Version 16.2* [Online]. Available: http://iphome.hhi.de/suehring/tml/download/old_jm/

[37] [Online]. Available: http://media.xiph.org/video/derf/

[38] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncell, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[39] *American National Standard for Telecommunications-Digital Transport of One-Way Video Signals-Parameters for Objective Performance Assessment*, ANSI Standard T1.801.03-2003, Jul. 2003.

[40] C. L. Phillipsa, J. A. Anderson, and S. C. Glotzer, "Pseudo-random number generation for Brownian Dynamics and Dissipative Particle Dynamics simulations on GPU devices," *J. Comput. Phys.*, vol. 230, no. 19, pp. 7191–7201, Aug. 2011.

Mehdi Fallahpour is currently a Post-Doctoral Fellow with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada. His experience is in multimedia security, watermarking, data hiding, and 3-D video. Prior to his current position with the University of Ottawa, he was a Post-Doctoral Fellow with the Communications Research Center Canada, Ottawa, where he was involved in a number of watermarking projects. His Ph.D. thesis, titled "High Capacity Data Hiding" was recognized as the sixth best Ph.D. thesis in the entire European Union in the field of security and trust management.

Shervin Shirmohammadi (SM'04) received the Ph.D. degree in electrical engineering from the University of Ottawa, Ottawa, ON, Canada.

He is currently a Full Professor with the School of Electrical Engineering and Computer Science, University of Ottawa. He is the Co-Director of the Distributed and Collaborative Virtual Environment Research Laboratory, and Multimedia Communications Research Laboratory, conducting research in multimedia systems and networking, specifically in gaming systems and virtual environments, video systems, and multimedia-assisted biomedical engineering. He has published more than 200 publications, over 12 patents and technology transfers to the private sector, and a number of awards and prizes.

Dr. Shirmohammadi is an Associate Editor-in-Chief of the IEEE Instrumentation and Measurement Magazine, Associate Editor of the IEEE Transactions on Instrumentation and Measurement, Senior Associate Editor of *ACM Transactions on Multimedia Computing, Communications, and Applications*, and was an Associate Editor of the *Springer's Journal of Multimedia Tools and Applications* from 2004 to 2012. He is a University of Ottawa Gold Medalist, a licensed Professional Engineer in Ontario, and a Lifetime Professional Member of the ACM.

Mehdi Semsarzadeh received the Ph.D. degree in computer engineering from the University of Tehran, Tehran, Iran, in 2013.

He is currently a Post-Doctoral Fellow with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada. His current research interests include video coding, video watermarking, and cloud gaming.

Jiying Zhao (M'00) received the Ph.D. degree in electrical engineering from North China Electric Power University, Beijing, China, and the Ph.D. degree in computer engineering from Keio University, Tokyo, Japan.

He is a Professor with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada. His current research interests include image and video processing and multimedia communications.

Dr. Zhao is a member of the Institute of Electronics, Information and Communication Engineers and the Professional Engineers Ontario.