Video-Tampering Detection and Content Reconstruction via Self-Embedding

Vahideh Amanipour and Shahrokh Ghaemmaghami, *Member, IEEE*

***Abstract*— Rapidly improving video-editing software tools and algorithms have made video content manipulation and modification feasible even by inexpert users. Detecting video tampering and recovering the original content of the tampered videos is, thus, a major need in many applications. Although detection and localization of the tampering in certain types of video editing have successfully been addressed in the literature, attempts for recovering the tampered videos are bound to methods using watermarks. In this paper, a scheme for the reconstruction of the tampered video through watermarking is proposed. The watermark payload, which consists of highly compressed versions of keyframes of the video and localization information, is embedded in the video using fountain coding. In addition to tampering localization, the proposed scheme can subsequently recover the content of the original video that has undergone malicious attacks. The proposed reconstruction method is quite capable of recovering the video for tampering rate as high as 67% and outperforms the latest tampered video recovery schemes in terms of both video quality and tampering percentage.**

***Index Terms*— Fountain coding, self-embedding, video watermarking, video-tampering detection.**

I. Introduction

EDITING video content has become increasingly simple, since the past decade and powerful video-editing technology is now available to the public. Video content is thus vulnerable to malicious attacks that can efficiently and seamlessly alter the video content. Detecting the editing history of the multimedia and reconstructing the content is among the most compelling features of multimedia authentication, especially in video surveillance, forensics, law enforcement, and content ownership. For instance, the authenticity and reliability of the CCTV-recorded video, which is presented as an evidence in law enforcement applications, is of great importance.

Malicious attacks and video tampering may generally be categorized into two classes.

1) *Temporal Tampering:* In temporal or interframe editing, manipulation is applied to the sequence of frames, mainly affecting the time sequence of visual information captured by video-recording devices. The common attacks of this type are frame addition, frame removal, and frame reordering or shuffling.
2) *Spatial Tampering:* Malicious modifications change the content of a single frame or several frames in spatial or intraframe editing. The most common spatial-tampering attacks may be grouped into object removal, object addition, and object modification attacks.

The detection of such manipulations is performed through two general categories of authentication schemes: with watermarking and using no watermark (e.g., using footprints of tampering). Moreover, the detection scheme may just target the authentication of the multimedia, or may also localize the tampered part of the multimedia. Reconstruction of the original video content is, however, the most desired and significant outcome of a tampering-detection scheme.

In reconstruction schemes, the encoder embeds the *reference* (which describes the essential content of multimedia), together with content hashes for authentication and localization purposes, in the multimedia. This reference is used by the dedicated decoder to reconstruct a recovered version of the original multimedia (of lower quality). Given the localized information about possible tampering and comparing the multimedia with the recovered version of the original multimedia, one obtains significant information about the edition history of the multimedia. Such authentication and reconstruction schemes, thus, carry the term *self-embedding*. The term highlights the embedding of the reference in the multimedia. The embedding process may be performed at the time of video recording by the recorder, e.g., the video camera. Embedding the watermark should not affect the quality of the video, and the transparency of the recorded video should be preserved.

In this paper, we introduce a content-based video authentication and reconstruction scheme. The main idea is to view the video as a communication channel that is used to communicate the embedded watermark (that contains significant information about the original video). The scheme is based on self-embedding of the fountain-coded keyframes. Moreover, some basic information about the location of keyframes in the video is embedded in other frames. If the video frames contain sufficient authentic blocks, a fountain decoder may be employed to reconstruct the references of the keyframes and identify their original location in the video-frame sequence. It is then possible to recognize malicious attacks and manipulations, including deleting or adding a subset of frames and insertion or deletion of an object in a particular part of the video, using no extra metadata such as frame headers.

The proposed scheme is successful in recovering the original keyframes of a tampered video, even when the tampering rate is as high as 67%. Moreover, the quality of the recovered video

Manuscript received March 10, 2017; revised September 2, 2017; accepted September 28, 2017. Date of publication December 27, 2017; date of current version February 8, 2018. The Associate Editor coordinating the review process was Dr. Shutao Li. *(Corresponding author: Shahrokh Ghaemmaghami.)*

V. Amanipour is with the Department of Electrical Engineering, Sharif University of Technology, Tehran 14588, Iran (e-mail: amanipour@ee.sharif.edu).

S. Ghaemmaghami is with the Electronics Research Institute, Sharif University of Technology, Tehran 14588, Iran (e-mail: ghaemmagh@sharif.edu).

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.2017.2777620

is higher than that of the existing video-recovery methods. When the tampering rate exceeds 67%, the tampering is localized, although the original keyframes may no longer be reconstructed.

This paper is organized as follows. We review the related work on video-tampering detection and authentication with and without using watermarks, as well as image content reconstruction using self-embedding, in Section II. Section III describes the proposed encoding scheme, which consists of an appropriate keyframe selection and the self-embedding of fountain-coded references. The corresponding decoding scheme, which consists of interframe and intraframe modules, is then discussed in Section IV. Experimental results are given in Section V, and conclusions and future work appear in Section VI.

II. Related Work

Video-tampering and editing-detection schemes may be categorized into two general types, depending on whether they use watermarking or not.

A. Editing Detection Without Watermark

Detection of editing history using no watermark or passive-tampering detection may be realized through the following major attempts [1]:

1) camera-based editing detection;
2) detection based on coding artifacts;
3) detection based on inconsistencies in the content;
4) copy–move detection in videos.

Each one of the aforementioned attempts detects a specific type of tampering under some restricted conditions. If the conditions are not satisfied, the detection of tampering fails. Moreover, it is not possible to recover the original content of the video through any of these methods. For instance, coding artifacts are used in [2] to detect frame deletion or insertion. In this paper, the video is first encoded using a fixed GOP encoder, and after frame insertion or deletion, it is reencoded. Among the limitations of this method are the failure to locate the exact position of frame insertion or removal. Moreover, the detection fails when the encoder is not a fixed GOP. Another sample of editing detection without watermark (of type 4) is found in [3]. The proposed scheme detects tampering when a spatiotemporal region of a sequence is replaced by either a series of fixed images (repeated in time) or a portion of the same video (taken from a potentially different time interval). This scheme fails to detect tampering, if the replaced frame sequence belongs to a different video.

B. Editing Detection With Watermark

Watermark embedding is used to strengthen the detection of malicious modifications of the video content. There are three main approaches in editing detection using watermark embedding, or active tampering detection.

1) *Authentication methods*, which detect whether the video is modified or not without giving any information about the location and type of the tampering. Appropriate features of the video as a whole are extracted and embedded as a watermark in such authentication schemes. These features are then extracted and compared with the corresponding features of the suspicious video and are subsequently used to characterize the video as tampered or nontampered. The selected features should be robust to typical editing procedures of the video, such as video compression and video scaling. For instance, in [4], local relative correlations are embedded as watermarks and classification of the video into tampered and nontampered classes is performed by measuring the distance vector between local relative correlation and embedded local relative correlation. Another such authentication scheme is introduced in [5].
2) *Localization methods*, which locate the tampered part of the video, in temporal or spatial domains, as well as the authenticity of the video. The features of the video, embedded as the watermark in such schemes, have a local nature that yields information about the tampered locations. For instance, the detection technique based on shot segmentation is proposed in [6] to locate the tampering in both spatial and temporal domains and the localization method in the temporal domain is used in [7]. Moreover, in [8], macroblock and frame indices are embedded into the nonzero quantized discrete cosine transform (DCT) values of blocks, enabling the detection of spatiotemporal tampering. Nonmalicious alterations made to the video, due to distortion, recompression, filtering, and so on, can be distinguished from malicious attacks by observing the macroblocks whose indices are not extracted correctly. When the change made to the video is nonmalicious, such incorrectly indexed macroblocks are randomly spread over the frames. But in the cases of malicious attacks, these macroblocks appear continuously in the same place in a series of frames. As other samples of such localization schemes, one can refer to [9] and [10].
3) *Content-reconstruction methods*, which not only determine the type and location of tampering but also recover the original content. In the center of any reconstruction scheme is the embedding of a *reference*, which describes the essential content of the video. Such schemes are studied under the label *self-embedding* in the literature. The embedded reference is used by the decoder to construct a recovered version of the original video whenever any form of tampering is detected. For instance, such self-embedding schemes are proposed in [11]–[14].

The advantage of self-embedding content-reconstruction schemes over ordinary authentication and localization schemes is quite clear. In most applications, including the law enforcement applications mentioned in Section I, access to the original video (or other multimedia) is of utmost importance. Although extensive research is devoted to the study of reconstruction schemes for images, there are few papers on video-reconstruction schemes. We leave the comparison of our proposed scheme with some of the existing methods to Section V on experimental evaluations.

C. Self-Embedding and Reconstruction of Image

Content-reconstruction schemes for images are thoroughly studied in the literature and several methods are proposed for tampering detection and recovery of the original image [15]–[22]. An interesting approach is presented in [15], where content reconstruction in self-embedding systems is viewed as dealing with an erasure communication channel. A reference that is constructed from the original image is coded and self-embedded in the image. For fixed (relatively large) tampering rate, it is shown that the reconstruction quality of the scheme proposed in [15] is significantly better than those given in [16]–[19].

A scheme for tampering detection and recovery for image is proposed in [20]. The half-tone image is used as the watermark and is self-embedded using quantization index modulation (QIM). Tampering is detected by extracting this reference and comparing it with the image using the Structural SIMilarity (SSIM) measure. A frame-by-frame application of the method to a video sequence is then used to give a tampering detection and recovery scheme for videos.

In [21] and [22], methods for tampering detection and recovery of region of interest (ROI) for medical images is proposed, which employs reversible watermarking. When there is no tampering, the original image is exactly reconstructed from the watermarked one, while for tampered ROI, the tampered area is localized and recovered losslessly.

Inspired by the success of [15] in the context of image recovery, we propose a content-reconstruction scheme for video based on self-embedding of the fountain-coded reference. It is of course possible to view a video as a sequence of images, and use the existing image content-reconstruction methods. For instance, this is the method used in [23] and [20]. However, this approach cannot detect temporal tampering and could be far from efficient because of its large computational load. Instead, we introduce a more practical reconstruction method via localization and self-embedding of the fountain-coded reference in an appropriate and effective way. The scheme recovers the keyframes of the original video with relatively high quality and obtains information about the authenticity and editing history of the tested video. While the performance becomes better in comparison with the existing methods for low tampering rates, the main significance of our method shows up in high-quality reconstruction of the original content for highly tampered video sequences. This point is further discussed in Section V-E.

III. Encoder

We assume throughout this paper that the editing history of the given video consists of inserting/removing objects into/from frames, removing a subset of adjacent frames, adding a sequence of frames from the same video or a different video, shuffling (reindexing) frames, changing the frame rate, or a combination of these modifications. Such modifications may or may not be malicious. Nevertheless, we would ideally like to identify the editing history.

The generic system diagram of the encoder in our proposed scheme is shown in Fig. 1. The process consists of four steps.

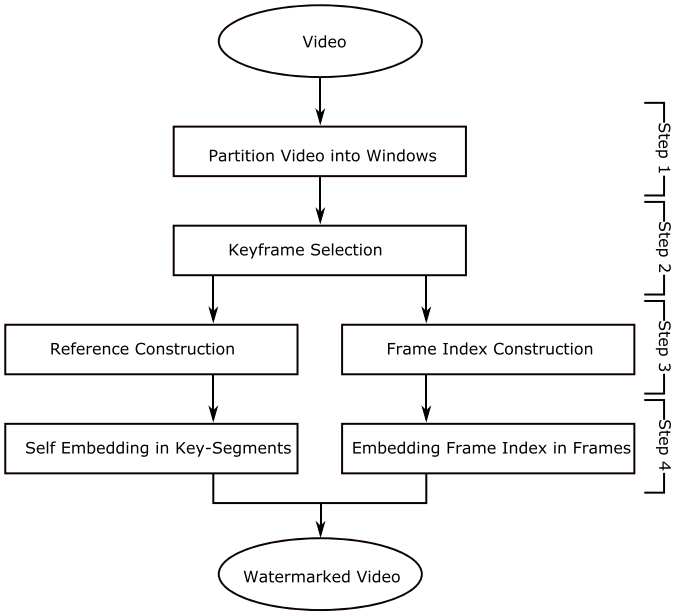


Fig. 1. System diagram of the proposed encoder.

Step 1 is to partition the video into nonoverlapping packages of w_ℓ frames. Each such package will be called a *window of frames*. In practice, the length w_ℓ should be small enough for real-time encoding and large enough so that the encoding process is efficient. If we select the length w_ℓ so that it is close to the number of frames in one second of the video signal (i.e., the frame rate), we cover all the visual events lasting around 1 s. In Step 2, a keyframe (with frame number k) is chosen in each window of frames and a *key-segment* of length $2G+1$ is identified around it. Here, G is a fixed chosen value. We call the union of frames corresponding to the numbers

$$k-G, k-G+1, \ldots, k-1, k, k+1, \ldots, k+G$$

the key-segment for the window of frames (see Fig. 2). The keyframe itself is, by definition, the center frame in the key-segment. The length $2G+1$ of the key-segment should be small relative to the window length w_ℓ. The details of keyframe selection appear in Section III-A. In Step 3, a reference is constructed from the keyframe, which records the content of the keyframes in a compressed form. Moreover, for every frame in the window, which is not in the key-segment, we form the *frame index* by including the window number, the frame number in the window, and the distance between the frame and the keyframe in the same window. The constructions of the reference and frame index appear in Section III-B. In Step 4, the coded reference is embedded in the key-segment. Moreover, the frame index for each frame, which is not in the key-segment, is embedded in that same frame. The details of self-embedding and coding the reference appear in Section III-C.

The keyframes are viewed as the skeleton of the video in this paper. The intraframe modifications may be detected once authentic keyframes are recovered. On the other hand, interframe editing (which includes insertion, removal, and replacement of the frames, and changes in the frame rate) is detected using the frame indices.

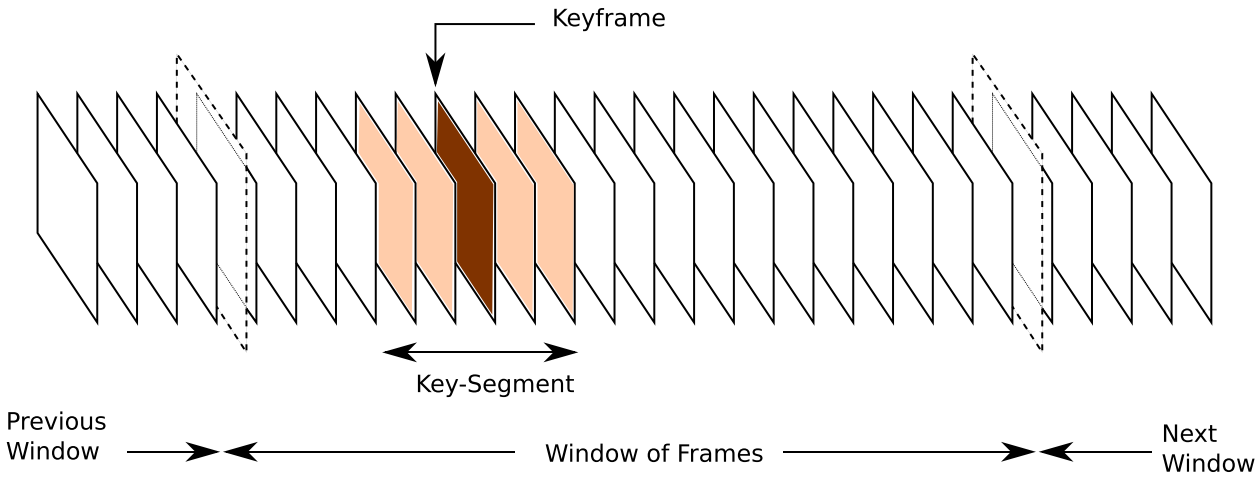


Fig. 2. Keyframe and key-segment in a window of frames.

A. Keyframe Selection

The choice of the keyframe is important in the encoding scheme, since the content of the video should be represented by the sequence of the constructed keyframes. There are many algorithms available in the literature to select the keyframes [24]. In this paper, selection is based on the shot-change-boundary positions in the frame window, using histogram comparison methods [25].

For each frame j, the dissimilarity feature value

$$d(j) = \sqrt{\sum_{b=1}^{M} (\text{Hist}_{j-1}(b) - \text{Hist}_j(b))^2} \tag{1}$$

is calculated, where Hist_j denotes the intensity histogram of the frame j and b is one of the M possible intensity values. The keyframe k is chosen, so that $d(k) \geq d(j)$ for any other frame j in the frame window. If the distance of the keyframe from the beginning frame of the frame window is less than G (i.e., if $k < G$), we replace k by G. Similarly, if the distance of the keyframe from the last frame of the frame window is less than G (i.e., if $k > w_\ell - G$), we replace k by $w_\ell - G$.

The frames within a frame window are expected to be almost similar and each such frame can be a representative of the window's content. However, the above choice guarantees that local dissimilarities, such as shot boundaries, are also effectively reflected in the sequence of keyframes.

B. Reference and Frame-Index Construction

Since the embedding capacity of the video frames is limited, the size of the constructed reference should be as small as possible. For this purpose, we use a highly compressed version of the selected keyframes as the content reference. An alternative method, which is also popular in the literature, is applying binarization techniques, which results in a binary image demanding only 1 b/pixel. Samples of such binarization techniques include dithering, thresholding, and edge detection [26], [27].

The compression method of [28] is used for constructing the content reference R. The grayscale keyframe is first decomposed into 8×8 blocks. Depending on the number of edge points in each block, we classify the blocks into four types [Fig. 3(a)–(d)]. The DCT is then applied to each block and the DCT coefficients are quantized using the standard JPEG quantization table. The corresponding bit allocation of the quantized coefficient for each type is determined according to the tables in Fig. 3. It is to be noted that the dc coefficient is converted into unsigned binary bits, while ac coefficients are converted into signed binary bits. The above compression method assigns 80, 60, 40, and 22 bits to blocks of Fig. 3(a)–(d), respectively. Two additional bits are used for each block to record the type of the block. An average of 52.2 b/block is thus used in this method.

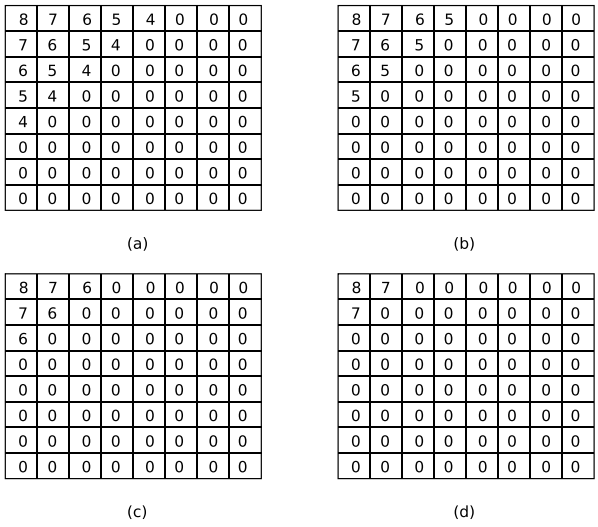


Fig. 3. Bit allocation tables. (a) Block type 1. (b) Block type 2. (c) Block type 3. (d) Block type 4.

The frame index $I_j = (nw_j, j, dk_j)$ for a frame j consists of the following information:

1) the number nw_j of the window containing the frame j;
2) the frame number j within its frame window;
3) the distance dk_j from frame j to the keyframe k in the same window, which is between $G - w_\ell$ and $w_\ell - G$.

C. Self-Embedding of Reference and Frame Index

Communication of the reference to the decoder may be viewed as a communication erasure channel. The references form the information that should be transmitted through the aforementioned communication channel. Each reference R is viewed as a sequence $R_1, \ldots, R_n$, where every R_i is a message of SL bits (where SL stands for symbol length) and n is the number of input symbols.

Each frame in the key-segment is decomposed to N blocks of size $b \times b$. The size of these blocks indicates the precision of the tamper localization. If W denotes the video width and H denotes the video height, we obtain

$$N = \frac{WH}{b^2}. \tag{2}$$

For each frame j, these blocks are denoted by $B_{j,1}, \ldots, B_{j,N}$. The reference, after fountain coding, is embedded in the blocks

$$B_{j,p}, \quad 1 \le p \le N \text{ and } k - G \le j \le k + G.$$

In other words, each one of the above $(2G + 1)N$ blocks carries a symbol of the watermark payload that is obtained by fountain coding from $R_1, \ldots, R_n$. The corresponding payload for $B_{j,p}$ will be denoted by $W_{j,p}$ and is computed as a pseudorandom linear combination of $R_1, \ldots, R_n$. Thus the symbol length of $W_{j,p}$ is equal to SL. Using the pseudo-random linear combination is due to the method employed in random linear fountain (RLF) codes. Instead, we may also use Luby transform or Raptor (Rapid Tornado) codes for a better performance, as discussed below. The experimental results of this paper are, however, based on the RLF scheme.

We also embed the authentication information in each block. The fountain decoder sees the *transmitted* symbol either as correctly transmitted or erased for authentic and tampered blocks, respectively. The authentication information is a hash function of $B_{j,p}$ and $W_{j,p}$, i.e.,

$$H_{j,p} = \text{Hash}(B_{j,p}, W_{j,p}, j, p, \text{key}). \tag{3}$$

The information embedded in $B_{j,p}$ consists of $(W_{j,p}, H_{j,p})$ if j belongs to the key-segment. For frame j, which is not in the key-segment, the information embedded in the blocks $B_{j,p}$ is $I_j = (nw_j, j, dk_j)$. The embedding scheme used for such frames is not necessarily the same as the self-embedding scheme used for the frames in the key-segment.

For the embedding algorithms, there are a couple of issues that should be considered.

1) *Imperceptibility:* The watermark embedding should not decrease the video quality.
2) *Capacity:* For fixed tampering rate, the higher the embedding capacity of a frame is, the larger the size of the reference can be chosen [this appears in (7)]. On the other hand, since embedding both the reference and the hash requires large capacity, most proposed schemes in the literature for image self-embedding use 3LSB for embedding, either in the luminance component Y (i.e., 3 b/pixel) or in the red, green, and blue (RGB) bit planes (i.e., 9 b/pixel).
3) *Robustness:* The watermark may be chosen robust or fragile against video processing, such as compression.

These three aspects of watermarking schemes are essentially interrelated. For instance, increasing the embedding capacity will decrease the imperceptibility and robustness. However, for embedding the frame index, the embedding capacity is not an issue, since the size of the frame index is negligible compared with the reference. The only relevant issues in the embedding of the frame index are, thus, the imperceptibility and robustness.

The frame index for every frame in the key-segment may also be embedded in the corresponding frame. If we choose to do so, the embedding schemes used for frame-index embedding and reference embedding should not interfere. For example, the frame index may be embedded in the first LSB, while the reference is embedded in the second LSB.

D. Analysis of Tampering Rate

The design of the fountain codes implies that if at least $Cn\log(n)$ of the $(2G + 1)N$ blocks survive the malicious attacks (where C is a constant depending on the required reliability of the scheme), the reference may be recovered from the remaining authentic blocks [29]. It is thus best to make sure that $(2G+1)N$ is large compared with $Cn\log(n)$. If s_{ref} denotes the size of the reference and c_{frame} denotes the pure embedding capacity of a single frame (i.e., the total embedding capacity minus the capacity devoted to the hash embedding), we set $r_{\text{frame}} = ((c_{\text{frame}})/(s_{\text{ref}}))$ and conclude

$$\begin{aligned} SL &= \frac{s_{\text{ref}}}{n} = \frac{c_{\text{frame}}}{N} \\ \Rightarrow n &= N\frac{s_{\text{ref}}}{c_{\text{frame}}} = \frac{N}{r_{\text{frame}}}. \end{aligned} \tag{4}$$

We then need to make sure that

$$\begin{aligned} (2G+1)N &\ge Cn\log(n) \\ &= C(N/r_{\text{frame}})(\log(N) - \log(r_{\text{frame}})) \\ \Leftrightarrow N &\le r_{\text{frame}} \exp\left(\frac{(2G+1)r_{\text{frame}}}{C}\right). \end{aligned} \tag{5}$$

The number of authentic blocks needed for the reconstruction of the reference should be at least greater than a threshold which may be computed (with a similar argument) as

$$T_{\text{frame}} = \frac{C \cdot N}{r_{\text{frame}}} \log\left(\frac{N}{r_{\text{frame}}}\right). \tag{6}$$

Since the tampering rate TR (for which content reconstruction is possible) does not exceed $1 - ((T_{\text{frame}})/((2G + 1)N))$, we find

$$\begin{aligned} TR &\le 1 - \frac{CN}{(2G+1)Nr_{\text{frame}}} \log\left(\frac{N}{r_{\text{frame}}}\right) \\ &= 1 - \frac{C}{(2G+1)r_{\text{frame}}} \log\left(\frac{N}{r_{\text{frame}}}\right) = \rho. \end{aligned} \tag{7}$$

In our proposed scheme, the final inequalities in (5) and (7) are satisfied and the corresponding tampering rates are reasonable.

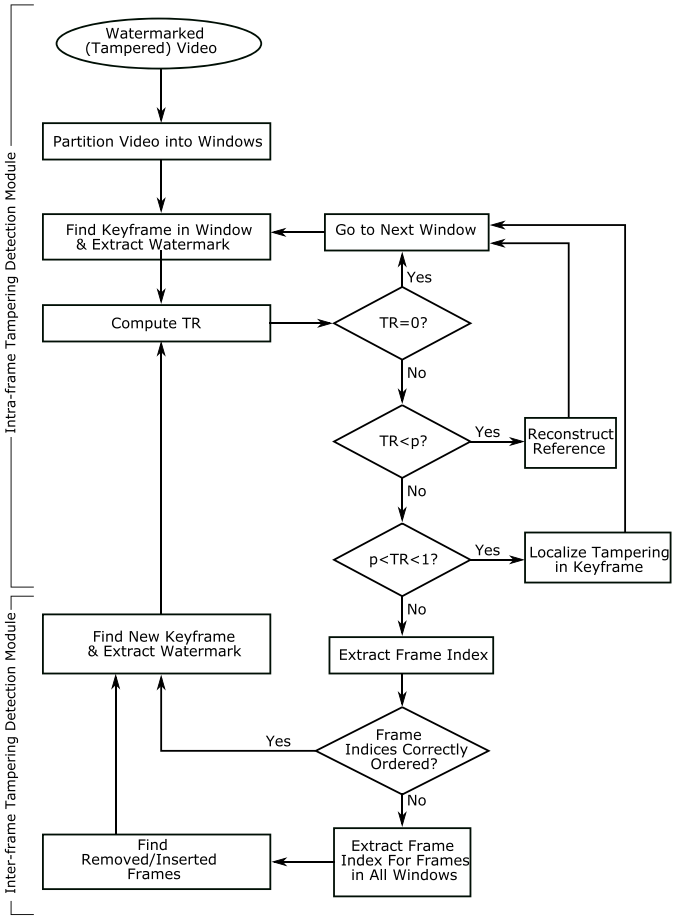


Fig. 4. System diagram of the proposed decoder.

IV. Decoder

The system diagram of the decoder in our proposed authentication scheme is illustrated in Fig. 4. The decoder scheme may be considered as a union of intraframe and interframe tampering-detection modules.

The intraframe tampering-detection module, which operates first, is pretty fast. In this module, the video is first partitioned into windows of length w_ℓ. In the initial window, the keyframe k is determined as described in the encoder scheme and the embedded watermark $(W'_{k,p}, H'_{k,p})$ is extracted. For $p = 1, \ldots N$, if the extracted $H'_{k,p}$ is equal to

$$\widetilde{H}_{k,p} = \text{Hash}\big(\widetilde{B}_{k,p}, W'_{k,p}, k, p, \text{key}\big) \tag{8}$$

the block $\widetilde{B}_{k,p}$ in the keyframe k, which corresponds to p, is declared authentic and we set $e_{k,p} = 0$. Otherwise, we set $e_{k,p} = 1$ and the tampering rate for the frame k is computed as

$$TR = \frac{1}{N}\sum_{p=1}^{N} e_{k,p}. \tag{9}$$

The error map of the frame k is defined to be the set

$$\{e_{k,p} \mid p = 1, \ldots, N\}. \tag{10}$$

If all blocks in the keyframe k are declared authentic, we move to the next window and repeat the above steps.

If the tampering rate is below the threshold ρ that was discussed in (7) (i.e., if TR$<\rho$), the recovery is possible. In this case, the reference for the keyframe is recovered by extracting the watermarks embedded in the key-segment. The blocks in the keyframe, which are not declared authentic, are then replaced by the corresponding blocks in the reconstruction reference.

If $\rho < TR < 1$, the self-recovery is no longer possible and the decoder can only localize the tampering. The error map indicates the location of the tampered blocks in this case. In case TR $= 1$, it is possible that the keyframe is incorrectly selected due to tampering. In this case, the frame index is extracted for the frames in the same window. If the extracted frame indices indicate that the frames belong to the chosen window and are correctly ordered, the frame number for the new keyframe number is determined and the above steps are repeated. Otherwise, the decoder realizes that interframe tampering has occurred and follows the forthcoming interframe tampering-detection module.

The interframe tampering-detection module first extracts the frame indices for all frames starting from the last authenticated keyframe. Using the extracted indices, the inserted or removed frames are distinguished and the location of the new keyframes is determined. The authentication and recovery process is then repeated using these new keyframes.

It is to be noted that when the video is not tampered, or when only intraframe tampering is performed over the video, the scheme is quite fast, since we only use the intraframe tampering-detection module.

V. Experimental Evaluation

The ten test videos used in the experimental evaluation come from the REWIND video copy–move forgeries database [33]. Each sequence has a resolution of 240×320 pixels and a frame rate of 30 frames/s. The length w_ℓ of the window is set equal to 20 frames. We are, thus, able to cover all visual events lasting around 0.67 s or more. The video frames are divided into $N = 1200$ blocks of size 8×8. The frame indices and the references of the keyframes are embedded in the second LSB of the luminance component Y. The capacity of embedding is, thus, equal to 64 bpb (bits per block) or, equivalently, 1 bpp (bit per pixel).

Choose the length $2G + 1$ of the key-segment equal to 5 or, equivalently, set $G = 2$. In each 8×8 block in the key-segment, 32 bits are used for hash embedding. The hash is generated by the MD5 algorithm [34]. The symbol length SL is also set equal to 32, i.e., 32 bits are dedicated to the embedding of the fountain-coded reference in each block within the key-segment.

The size of each frame index I_j is set equal to 24 bits, where 14 bits are dedicated to window number nw_j, 5 bits are allocated to the frame number j in the window, and 5 bits are used for the distance dk_j of j from the keyframe in the same window. The frame index I_j is embedded in each 8×8 block of the frame j twice (thus, the embedding capacity of each block is 48 bits).

The decoder extracts the frame indices $I'_{j,2p-1}$ and $I'_{j,2p}$ from each block $B_{j,p}$. We choose the *recognized frame index* I'_j so that the following holds.

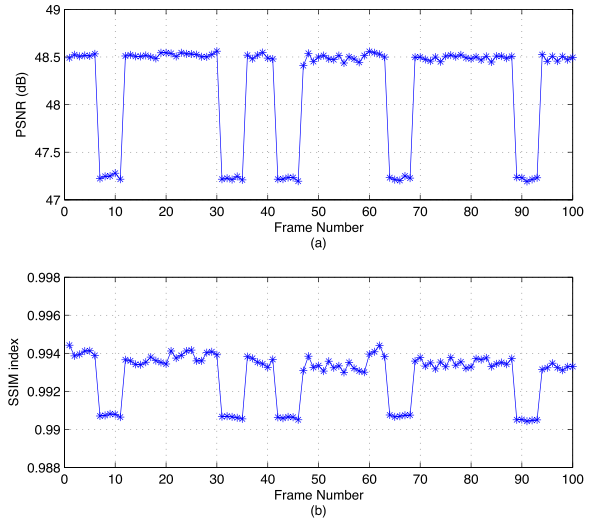


Fig. 5. Imperceptibility evaluation results for the measured (a) PSNR and (b) SSIM.

1) $\sigma(I'_j) = \#\{p \in \{1, 2, \ldots, 2N\} \mid I'_{j,p} = I'_j\}$ is maximum.
2) $\sigma(I'_j) > T(N)$, where $T(N)$ denotes a threshold depending on N.

In our experiments, we set $T(N) = (N/10) = 120$, i.e., the recognized frame index needs a minimum of 120 votes to be confirmed.

A. Imperceptibility Evaluation

For imperceptibility evaluation, we compare the qualities of the watermarked and the original frames using the peak signal-to-noise ratio (PSNR) and the SSIM index [35]. While PSNR is very popular and widely used to measure visual distortion, it is just an overall indication of the visual quality of an image or a video signal. Accordingly, the PSNR may have poor correlation with human visual perception in some conditions, particularly when the distortion is spatially and/or temporally localized. Therefore, in addition to the PSNR, the SSIM index is calculated in this paper. The SSIM index is a number between 0 and 1, where 0 shows zero correlation between frames, and 1 indicates that they are identical [35].

In Fig. 5, the PSNR and SSIM index for 100 frames from one of the test videos are presented. When we embed in the second LSB, if distribution of data in the three LSB is uniform, the PSNR is equal to 45.1 dB, independent of the video chosen. Moreover, embedding in the third LSB decreases the PSNR to 39.1 dB. In order to improve the quality of the watermarked frame, a smoothing function is employed to minimize the difference between the original pixel and the corresponding pixel in the watermarked frame. This function modifies the first LSB depending on the value of the second LSB and the watermark. So the PSNR is increased by more than 2 dB.

One can observe from Fig. 5(a) that the PSNR for the key-segment is around 47.2 dB, while the PSNR for the rest of frames is around 48.5 dB. Note that 48 bpb is used for frame-index embedding, while 64 bpb is used for reference embedding in the key-segment. This means that the embedding capacity used for nonkey-segment frames is around $48/64 = 75\%$ of the embedding capacity used for the key-segment (and this is the reason for the difference between the corresponding PSNRs). It is to be noted that the quality of the image is good enough for most applications, if the value of the PSNR is above 35 dB. For instance, the normally accepted PSNR quality is 35 dB in the evaluation of watermarking transparency [30], forensics and law enforcement applications [31], and medical video sequences [32].

The average value of the SSIM index for the key-segments (in comparison with the corresponding original frames) is equal to 0.9906 and for the rest of frames is equal to 0.9935. These values are far beyond the human visual system threshold of noticeable distortion.

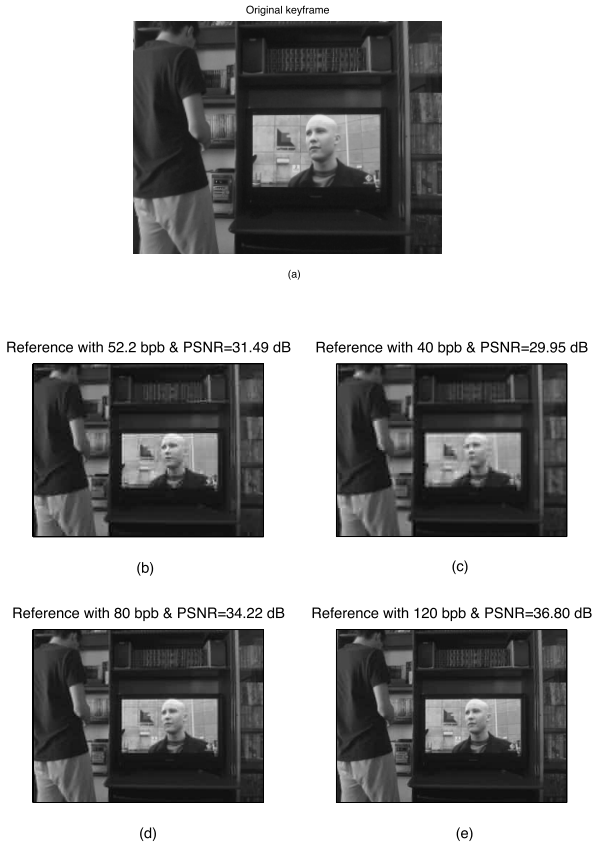


Fig. 6. (a) Original keyframe. Reference with (b) 52.2, (c) 40, (d) 80, and (e) 120 bpb.

B. Reference-Quality Evaluation

Fig. 6(a) presents the keyframe. The corresponding reference of different sizes is presented in Fig. 6(b)–(e), using 52.5, 40, 80, and 120 bpb, respectively. The reference in Fig. 6(b), which has the PSNR equal to 31.49 dB with respect to the original keyframe, is the one embedded in the video by the proposed algorithm, while the other references are included to illustrate the correlation between the embedding capacity and the quality of the reference. With reference to (7), when the tampering rate TR is fixed, the ratio

$$r_{\text{frame}} = \frac{c_{\text{frame}}}{s_{\text{ref}}} \tag{11}$$

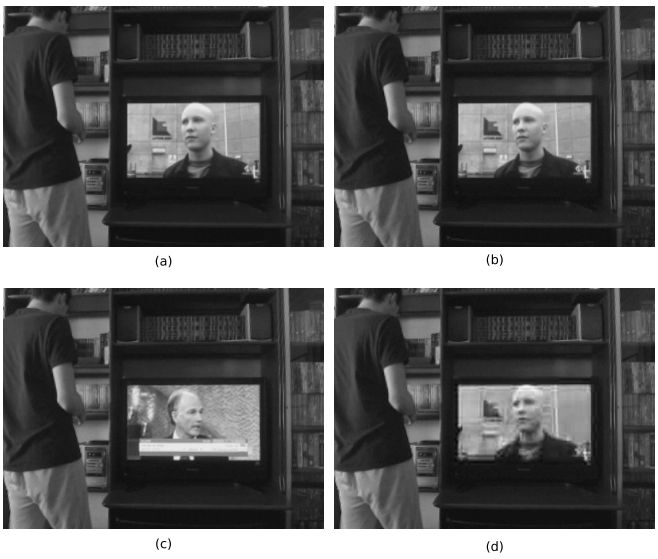


Fig. 7. (a) Original keyframe. (b) Watermarked keyframe with PSNR = 47.24 dB. (c) Tampered keyframe with TR= 16.5%. (d) Recovered keyframe with PSNR = 34.12 dB.

remains fixed as well. Consequently, if the embedding capacity is 1, 2, or 3 bpp (i.e., if the payload is embedded in 1, 2, or 3 LSBs), the size of the reference may be made one, two, or three times larger, respectively. The corresponding PSNRs of the references illustrated in Fig. 6(c)–(e) are equal to 29.95, 34.22, and 36.8 dB, respectively. As illustrated in this example, the quality of the reference improves by increasing the embedding capacity, for a fixed tampering rate.

C. Intraframe Tampering Detection and Recovery

In Fig. 7, the performance of the proposed scheme in recovering the content of the keyframe is illustrated. If the video undergoes intraframe tampering, the keyframe content changes. In this figure, the original keyframe, the watermarked keyframe, and the tampered keyframe (with a tampering rate of 16.5%) are shown in Fig. 7(a)–(c), respectively. Fig. 7(d) shows the recovered keyframe, where the tampered parts of the frame are replaced by the corresponding parts from the reconstruction reference. The content of the keyframe and the content of the key-segment are correlated and the tampering rate thus refers to the tampering rate in the key-segment. The alterations made to the content of the keyframe may clearly be realized from the recovered frame by comparing it with the tampered keyframe.

Experiments have been conducted to find out the highest rate of reversible tampering. Each line in Fig. 8(a) corresponds to a specific keyframe in one of the ten videos in the REWIND database [33]. The corresponding frame window undergoes tampering 20 times randomly with TR equal to $10\%, 20\%, \ldots, 60\%$ and 67%. The average PSNR (i.e., the quality) of the recovered keyframe is reported in the diagram. As expected, the average PSNR decreases as the tampering rate increases. The standard deviation of the computed values of the PSNR from the average PSNR decreases from 0.54 for TR = 10% to 0.13 for TR = 67%. In fact, the standard deviation is equal to 0.42 for TR = 20%, 0.28 for TR = 30%, 0.22 for TR = 40%, 0.19 for TR =50%, and 0.14 for TR = 60%. Fig. 8(b) illustrates the box plot of the data for the ten tested videos. The threshold $\rho = 67\%$, which also appears in 7, is independent of the video content and only depends on the length of the key-segment, embedding capacity, size of the reference, and the number of blocks in each frame. Recovering the keyframes fails when the tampering rate exceeds 67%, while tampering can still be located by the method. As explained in Section IV, the error map locates the tampered blocks when TR exceeds 67%, despite the failure of keyframe recovery. The average PSNR for the recovered keyframe varies from 41.6 to 33.2 dB, which is quite acceptable in most typical applications. Moreover, since the keyframe may be recovered for a tampering rate up to 67%, if less than four frames are removed from the $2G + 1 = 5$ key-segment frames, the keyframe may still be recovered.

TABLE I
TAMPER-DETECTION PERFORMANCE

Tests	TR	TP (%)	TN (%)
Video 01	26.0	100	98.2
Video 02	3.8	100	99.8
Video 03	3.2	100	99.9
Video 04	2.4	100	99.6
Video 05	15.2	100	99.9
Video 06	8.4	100	97.4
Video 07	17.1	100	97.3
Video 08	30.1	100	90.5
Video 09	0.5	100	100
Video 10	10.3	100	98.9
Average		100	98.1

To examine the performance of the tampering-detection scheme, the true-positive (TP) rate and the true-negative (TN) rate for the ten REWIND forged videos are reported in Table I. The TP rate is computed as the number of tampered blocks, which are correctly identified as tampered, normalized by the actual number of tampered blocks. The TN rate is computed as the number of blocks that are correctly identified as nontampered, normalized by the number of nontampered blocks. The reported TP and TN rates indicate that almost all tampered blocks are detected correctly by the proposed method, at least for the REWIND-forged video files used in the experiments. To have a basis for the comparison, note that the average (pixel based) TP and TN rates reported in [3] for the exact same set of videos are 75% and 97%, respectively. In the cases of FP, the reconstruction algorithm (like other existing algorithms in the literature) may degrade the

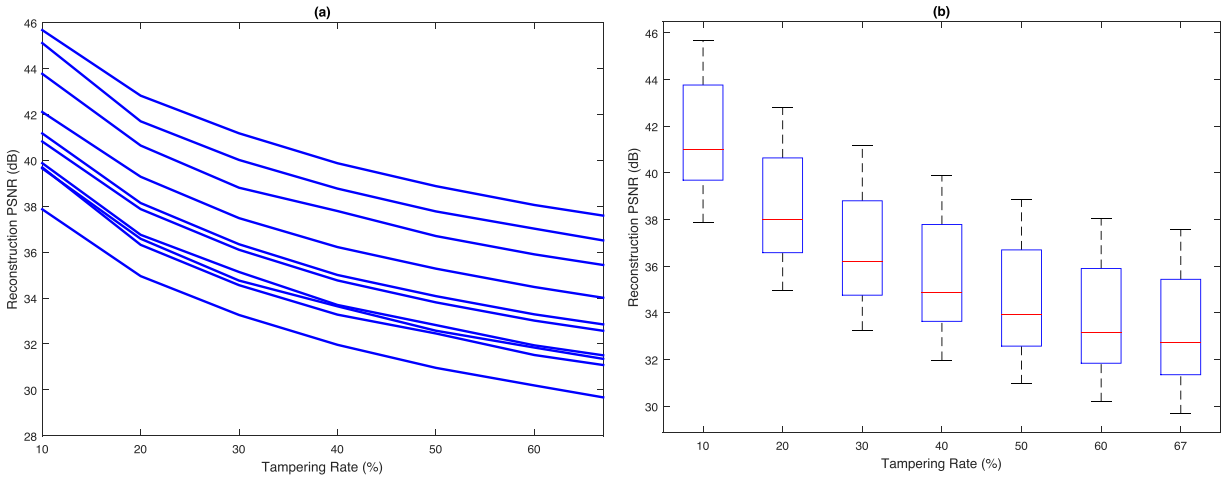


Fig. 8. Recovered keyframe PSNR versus tampering rate. (a) Average PSNR of the recovered keyframe after 20 random tamperings for ten different tested videos. (b) Box plot illustration of the same data.

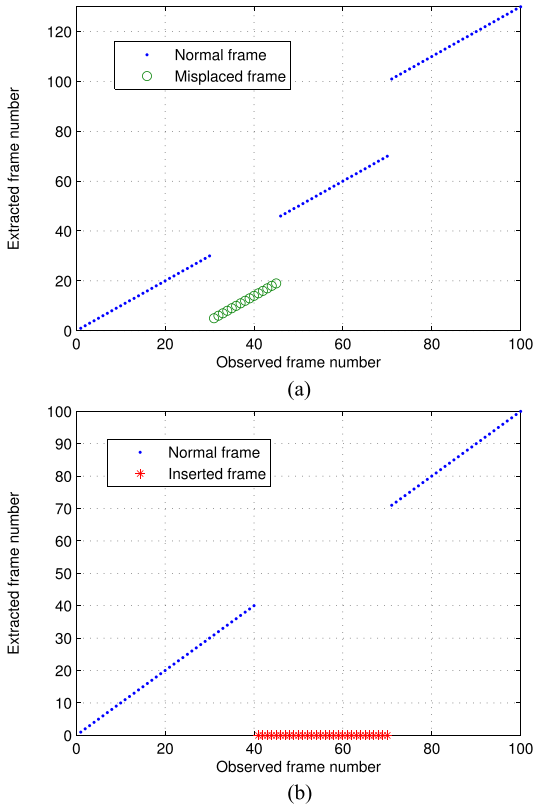


Fig. 9. Interframe tampering. (a) Removed frames and misplaced frames. (b) Inserted frames.

video quality. This, however, would affect a relatively small portion of the video, since the FP rate is small.

D. InterFrame Tampering Detection

In order to test the scheme against interframe tampering, frames 31–45 from a video consisting of 130 frames are replaced by frames 5–19 and frames 71–100 are removed from the video. Fig. 9(a) shows that the proposed method takes the tampered video as the input, and successfully extracts the correct frame number for frames 31–45 while realizing the removal of frames 71–100. In Fig. 9(b), the frames 41–70 of the given video have been replaced with 30 frames from another video. Extracting the frame indices fails to produce legitimate index sets in this case, and the aforementioned 30 frames are also not recognized as key-segments. Thus, the decoder successfully declares them as inserted frames. The original frame numbers for the frames in the key-segment are reconstructed from the information available in the frame indices for the rest of frames, and are presented in Fig. 9(a) and (b).

E. Comparison With Existing Methods

As mentioned earlier, most existing active video-tampering-detection methods in the literature only focus on the authentication of the video content or localization of video tampering. The number of methods proposed for the reconstruction of the content is relatively small. Nevertheless, several self-embedding content-reconstruction schemes have been proposed in the literature. For instance, Mobasseri and Evans [11] consider a pairing of the frames in the video. For each frame, 3 MSBs of the grayscale picture of it are embedded in the LSB planes of RGB of its couple (which is determined by pairing). Having tied one frame to another frame downstream, frame removal and insertion can be identified and reversed (subject to certain limitations). However, reindexing may damage the frame pair correspondences indicating a manipulation. Moreover, removal and insertion of frames at two different temporal neighborhoods completely impairs the synchronization of frame pairs, which are identified using a pseudorandom permutation matrix. This results in the loss of verification and self-recovery characteristics. Another content-reconstruction scheme is proposed in [12], which localizes the tampering and recovers the frame when the tampered area is limited, i.e., when the tampering rate is small. In particular, the quoted experimental results address tampering rates below 3%, which is in fact very small for actual applications. Moreover, since the frame indices are embedded in the

TABLE II
QUALITY OF THE RECONSTRUCTED FRAMES

TR	PSNR for proposed method	PSNR for [13]	PSNR for [14]
1.01%	53.04 dB	47.79 dB	42.44 dB
4.54%	49.47 dB	38.90 dB	36.45 dB
5.05%	35.76 dB	37.61 dB	36.52 dB
12.63%	40.60 dB	35.75 dB	32.48 dB
18.69%	38.71 dB	31.46 dB	27.53 dB

moving objects, the proposed scheme fails to authenticate the video when the frames do not contain any such objects, or when the moving objects are removed from the video frames.

The proposed scheme overcomes the aforementioned restrictions and makes it possible to reconstruct the video content when the tampering rate is as high as 67%. For comparison, it should be noted that the highest tampering rate for which successful reconstruction is reported in the references is less than 20%. When the tampering rate exceeds 67%, although tampering is localized by our scheme, the reconstruction of the tampered frames fails.

Hash functions and a nonnegative matrix factorization method are used in [13] to generate watermarks, both in block level and frame level. Such watermarks are then self-embedded into the video, so that the spatial and temporal tampering may be authenticated separately. The approximate recovery of the tampered features is based on the gray prediction method and spatial-temporal continuous characteristics of the video. The performance of the proposed scheme is then compared with the scheme proposed in [14] to provide evidence for the effectiveness of the method. The reconstruction scheme of [14] divides the frames into blocks and defines a signature and two *descriptions* for every such block. Either of the two descriptions (which have low and high quality) are enough to rebuild the altered block. The block descriptions and the signature are embedded using a doubly linked chain into the LSBs of distant blocks and the block itself.

Table II compares the PSNR of the frames recovered by our proposed scheme to the PSNR of the recoveries reported in [13] for a number of tampered videos. The second, third, and fourth columns of the table report the PSNR values using our method, the method in [13], and the method in [14], respectively. Except for one of the five tampering attacks, the quality of the recovered frame is significantly better when our proposed scheme is used. It is to be mentioned that the video sequences used by Hassan *et al.* [13] for comparisons, which are also used in Table II, do not have shot boundaries. Since the reconstruction scheme proposed in [13] relies on spatial-temporal continuous characteristics of the video, it is highly probable that the reconstruction fails when the shot boundaries of the video undergo tampering attacks. The advantage of our proposed scheme over the method in [13] would be more significant in such cases.

VI. CONCLUSION

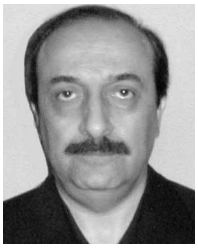
A video-content-reconstruction scheme has been proposed in this paper. We take the video as an erasure communication channel through which the watermarks, consisting of the compressed keyframes and localization data, should be transmitted. Fountain coding is employed to embed the watermark payload and extract them at the decoder end. The encoder consists of a keyframe selection module, a watermark construction module, and a watermark embedding module, while the decoder consists of an intraframe tampering-detection module (for detecting spatial modifications) and an interframe tampering-detection module (for detecting temporal modifications). The proposed method can localize the tampering and reconstruct the original keyframes with quality ranging from approximately 41.6–33.2 dB, when the tampering rate varies from 10% to 67%.

REFERENCES

[1] S. Milani *et al.*, "An overview on video forensics," *APSIPA Trans. Signal Inf. Process.*, vol. 1, pp. 1–18, Dec. 2012.
[2] A. Gironi, M. Fontani, T. Bianchi, A. Piva, and M. Barni, "A video forensic technique for detecting frame deletion and insertion," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)* May 2014, pp. 6226–6230.
[3] P. Bestagini, S. Milani, M. Tagliasacchi, and S. Tubaro, "Local tampering detection in video sequences," in *Proc. IEEE 15th Int. Workshop Multimedia Signal Process.*, Sep./Oct. 2013, pp. 488–493.
[4] R. Singh, M. Vatsa, S. K. Singh, and S. Upadhyay, "Integrating SVM classification with SVD watermarking for intelligent video authentication," *Telecommun. Syst.*, vol. 40, nos. 1–2, pp. 5–15, 2009.
[5] Q. Sun, D. 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.
[6] C.-Y. Liang, H. Wu, and A. Li, "Video content authentication technique based on invariant feature detection and cloud watermark," in *Proc. 8th Int. Conf. Intell. Syst. Design Appl.*, vol. 12. 2008, pp. 602–607.
[7] 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, pp. 413–426, Oct. 2011.
[8] M. Fallahpour, S. Shirmohammadi, M. Semsarzadeh, and J. Zhao, "Tampering detection in compressed digital video using watermarking," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 5, pp. 1057–1072, May 2014.
[9] C. H. Kung, P. T. Wu, and Y. C. Lee, "The design of an innovative method for digital video surveillance system with watermarking and error control codes," in *Proc. IEEE Instrum. Meas. Technol. Conf.*, May 2005, pp. 633–638.
[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. (I2MTC)*, May 2013, pp. 872–875.
[11] B. G. Mobasseri and A. T. Evans, "Content-dependent video authentication by self-watermarking in color space," *Proc. SPIE*, vol. 4314, pp. 35–44, Aug. 2001.
[12] Y. Shi, M. Qi, Y. Yi, M. Zhang, and J. Kong, "Object based dual watermarking for video authentication," *Optik-Int. J. Light Electron Opt.*, vol. 124, no. 19, pp. 3827–3834, 2013.
[13] A. M. Hassan, A. Al-Hamadi, Y. M. Y. Hasan, M. A. A. Wahab, and B. Michaelis, "Secure block-based video authentication with localization and self-recovery," *Int. J. Elect., Comput., Energetic, Electron. Commun. Eng.*, vol. 3, no. 9, pp. 69–74, 2009.
[14] M. Tong, J. Guo, S. Tao, and Y. Wu, "Independent detection and self-recovery video authentication mechanism using extended NMF with different sparseness constraints," *Multimedia Tools Appl.*, vol. 75, no. 13, pp. 8045–8069, 2016.
[15] P. Korus and A. Dziech, "Efficient method for content reconstruction with self-embedding," *IEEE Trans. Image Process.*, vol. 22, no. 3, pp. 1134–1147, Mar. 2013.
[16] X. Zhang, S. Wang, and G. Feng, "Fragile watermarking scheme with extensive content restoration capability," in *Proc. Int. Workshop Digit. Watermarking*, 2009, pp. 268–278.
[17] X. Zhang, S. Wang, Z. Qian, and G. Feng, "Reference sharing mechanism for watermark self-embedding," *IEEE Trans. Image Process.*, vol. 20, no. 2, pp. 485–495, Feb. 2011.

[18] X. Zhang, Z. Qian, Y. Ren, and G. Feng, "Watermarking with flexible self-recovery quality based on compressive sensing and compositive reconstruction," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 4, pp. 1223–1232, Dec. 2011.
[19] X. Zhang, S. Wang, Z. Qian, and G. Feng, "Self-embedding watermark with flexible restoration quality," *Multimedia Tools Appl.*, vol. 54, no. 2, pp. 385–395, 2011.
[20] P. G. Freitas, R. Rigoni, and M. C. Q. Farias, "Secure self-recovery watermarking scheme for error concealment and tampering detection," *J. Brazilian Comput. Soc.*, vol. 22, no. 1, p. 5, 2016.
[21] K.-H. Chiang, K.-C. Chang-Chien, R.-F. Chang, and H.-Y. Yen, "Tamper detection and restoring system for medical images using wavelet-based reversible data embedding," *J. Digit. Imag.*, vol. 21, no. 1, pp. 77–90, 2008.
[22] O. M. Al-Qershi and B. E. Khoo, "ROI-based tamper detection and recovery for medical images using reversible watermarking technique," in *Proc. IEEE Int. Conf. Inf. Theory Inf. Secur.*, Dec. 2010, pp. 151–155.
[23] A. Dziech *et al.*, "Overview of recent advances in CCTV processing chain in the INDECT and INSIGMA projects," in *Proc. IEEE Int. Conf. Availability, Rel. Secur. (ARES)*, Sep. 2013, pp. 836–843.
[24] R. W. Lienhart, "Comparison of automatic shot boundary detection algorithms," *Proc. SPIE*, vol. 3656, pp. 290–301, Dec. 1998.
[25] H. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Syst.*, vol. 1, no. 1, pp. 10–28, 1993.
[26] R. W. Floyd and L. Steinberg, "Adaptive algorithm for spatial greyscale," *Proc. Soc. Inf. Display*, vol. 17, no. 2, pp. 75–77, 1976.
[27] D. L. Lau and G. R. Arce, *Modern Digital Halftoning*. New York, NY, USA: Marcel Dekker, 2001.
[28] Z. Qian and G. Feng, "Inpainting assisted self recovery with decreased embedding data," *IEEE Signal Process. Lett.*, vol. 17, no. 11, pp. 929–932, Nov. 2010.
[29] D. J. C. MacKay, "Fountain codes," *IEE Proc.-Commun.*, vol. 152, no. 6, pp. 1062–1068, Dec. 2005.
[30] M. Hasnaoui and M. Mitrea, "Multi-symbol QIM video watermarking," *Signal Process., Image Commun.*, vol. 29, no. 1, pp. 107–127, 2014.
[31] X. Zhao and A. T. Ho, "Semi-fragile image watermarking, authentication and localization techniques for law enforcement applications," in *Handbook of Research on Computational Forensics, Digital Crime, and Investigation: Methods and Solutions*. Hershey, PA, USA: IGI Global, 2010.
[32] M. Razaak and M. G. Martini, "Rate-distortion and rate-quality performance analysis of HEVC compression of medical ultrasound videos," in *4th Int. Conf. Sel. Topics Mobile Wireless Netw. (MoWNet)*, vol. 40, 2014, pp. 230–236.
[33] *REWIND Video: Copy-Move Forgeries Dataset*. Accessed: Mar. 15, 2013. [Online]. Available: https://sites.google.com/site/rewindpolimi/downloads/datasets/video-copy-move-forgeries-dataset
[34] R. Rivest, *The MD5 Message-Digest Algorithm*, RFC Editor, USA, 1992.
[35] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

Vahideh Amanipour was born in Tehran, Iran, in 1983. She received the B.S. degree (Hons.) in electrical engineering from the K. N. Toosi University of Technology, Tehran, in 2005, and the M.S. degree in electrical engineering from the University of Tehran, Tehran, in 2007. She is currently pursuing the Ph.D. degree with the Sharif University of Technology, Tehran, where she is researching on video processing under the supervision of Prof. S. Ghaemmaghami.



Shahrokh Ghaemmaghami (M'95) received the B.S. and M.S. degrees from the Electrical Engineering Department, Sharif University of Technology (SUT), Tehran, Iran, and the Ph.D. degree from the Queensland University of Technology, Brisbane, QLD, Australia.

He is currently an Associate Professor of signal processing with SUT.

Dr. Ghaemmaghami has served as a member of technical committees and advisory boards of several international conferences and journals. He is a member of the Communications and System Engineering Group, Electrical Engineering Department and the Center of Excellence for Information Security, and the Director of the Electronics Research Institute, SUT. He is a member of the IEEE ComSoc, the IEEE Computer Society, and the ACM. He has led many large research projects in speech, image, and video processing and teaching courses in speech processing, including coding, recognition, and synthesis, and also in information hiding, e.g., watermarking, steganography, and steganalysis.