Signal Processing: Image Communication 102 (2022) 116593

Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

Imperceptible–visible watermarking for copyright protection of digital videos based on temporal codes

L. Velazquez-Garcia [a], A. Cedillo-Hernandez [b,*], M. Cedillo-Hernandez [c], M. Nakano-Miyatake [c], H. Perez-Meana [c]

[a] Instituto Politecnico Nacional, Centro de Investigaciones Económicas, Administrativas y Sociales, Lauro Aguirre 120, Agricultura, Miguel Hidalgo, C.P. 11360, Ciudad de Mexico, Mexico

[b] Instituto Tecnologico y de Estudios Superiores de Monterrey, Campus Hidalgo, Blvd. Felipe Angeles 2003, Venta Prieta, C.P. 42080, Pachuca de Soto, Hidalgo, Mexico

[c] Instituto Politecnico Nacional, SEPI ESIME Culhuacan, Avenida Santa Ana 951, San Francisco Culhuacan CTM V, Coyoacan, C.P. 04260, Ciudad de Mexico, Mexico

ARTICLE INFO

Keywords:
Video watermarking
Copyright protection
Imperceptible–visible video watermarking
Temporal codes

ABSTRACT

Digital watermarking has emerged as an important tool for copyright protection of digital videos. In this paper, we introduce a robust and imperceptible video watermarking method performed in the base-band domain that is not computationally expensive. Two nonoverlapped watermarks are embedded in the host video frames. The first watermark is a grayscale image that represents the ownership of the video. First, this watermark is decomposed into a sequence of binary images called temporal codes by using a spatiotemporal masking model. Then, the temporal codes are embedded in the spatial domain of video frames by using the imperceptible–visible watermarking paradigm. Since it is well known that such a paradigm is computationally expensive, we employ a technique that considers the video scene as the basic processing unit to considerably increase the execution speed. The second watermark works as a supporting media to share data between embedding and exhibition stages and is embedded in the discrete cosine transform (DCT) with a very robust method based on quantization index modulation (QIM) under dither modulation and some spatiotemporal criteria of the human visual system (HVS). Computer simulations were conducted regarding robustness, imperceptibility, and time consumption to determine the feasibility of our proposal. Experimental results confirm that the combination of temporal codes and the imperceptible–visible watermarking paradigm is an innovation in the field that brings advantages such as simplicity, low computational complexity, and improved robustness and imperceptibility.

1. Introduction

Due to the rapid development of information technologies, it is possible to create a perfect copy of a digital medium with high quality at a low cost and then distribute it on a large scale. Copyright ownership protection is necessary to safeguard digital media against unauthorized duplications and other illegal operations. Digital watermarking has emerged as a technology that aims to hide ownership information in a host signal in an invisible manner before distribution. All distributed copies containing the watermark can later be examined to establish ownership [1]. This paper focuses on the copyright protection of digital videos by using digital watermarking. Compared with still image watermarking, video watermarking introduces some specific requirements related to robustness, imperceptibility, and computational cost.

Regarding robustness, video watermarking faces aggressive intentional or unintentional operations that can remove the embedded watermark, such as transcoding and temporal desynchronization operations that are not present in image watermarking [2]. Imperceptibility metrics applied to measure video quality distortion must consider spatiotemporal aspects of the human visual system (HVS) to reflect the influence of quality alterations over time [3]. Additionally, the large volume of data in video sequences prevents a watermarking technique initially designed for still images from being applied to video without changes, which would require substantial computational costs. A well-designed video watermarking algorithm must take advantage of the redundancy between consecutive frames. The domain where the watermark embedding process is carried out impacts the robustness, imperceptibility, and computational cost of the video watermarking scheme. Some video watermarking techniques choose the compressed domain to embed and extract the watermarking signal [4–9]. This approach has two main advantages. The first advantage is practicality since the videos are usually stored in a compressed way. The second advantage is the low computational cost involved, which opens the option of generating real-time solutions. However, a large drawback is that any small change in the encoded information impacts the frame's visual quality. Therefore, very little information can be embedded.

* Corresponding author.
E-mail address: acedillo@tec.mx (A. Cedillo-Hernandez).

https://doi.org/10.1016/j.image.2021.116593
Received 30 June 2020; Received in revised form 6 December 2021; Accepted 8 December 2021
Available online 14 December 2021

Another disadvantage of these methods is that they are closely related to the video compression process, and the watermark often does not survive transcoding operations. Other video watermarking proposals use the base-band domain by modifying the frame pixels directly (spatial domain) [10–13] or by transforming them to frequency domains such as discrete cosine transform (DCT) and discrete wavelet transform (DWT) [14–17]. The most relevant advantage of these schemes is that several operations can be performed to improve robustness and imperceptibility. Additionally, as these methods are not associated with any specific video compression standard, the watermark is more likely to remain in the video after aggressive tasks, including transcoding, are performed. However, these techniques are computationally expensive because they usually perform frame-by-frame tasks along video sequences.

In this paper, we propose a robust video watermarking scheme performed in the base-band domain, with an imperceptible watermark at a low computational cost. The proposed watermarking scheme embeds two nonoverlapped watermarks in the host video frames. The first watermark is considered the most relevant within the proposed method. This watermark is a grayscale image that graphically represents the ownership data. This visible watermark is processed by being decomposed into a fixed number of binary images, called temporal codes, using the method proposed in [18]. The main objective of this procedure is to create n binary images through spatial and temporal masking effects to become less easily perceived, either in each video frame or by the temporal integration of the HVS. The way to recover the watermark signal is by performing pixelwise mathematical averaging of the video frames or by long-exposure photography [18]. A detailed review of the technique proposed in [18] is presented in Section 2 of this document. The temporal codes are embedded in the spatial domain of each video frame by using the *imperceptible–visible* watermarking paradigm to give robustness and imperceptibility to the proposed method. In a general way, this approach refers to embedding a visible watermark that is not perceptible to the naked eye. The watermark can be revealed only by performing image enhancement operations that are usually fast and noncomplex. The *imperceptible–visible* paradigm claims two advantages over conventional approaches. First, it allows ownership to be determined in a practical way (i.e., by avoiding complex detection stages), and second, it improves watermark imperceptibility. However, one of the main drawbacks of this approach is related to the high computational cost. The *imperceptible–visible* paradigm determines the watermark location by performing an exhaustive search of the host image's visual features, which becomes an issue for video watermarking. To address this problem, we design a procedure to greatly reduce the computational cost of the *imperceptible–visible* paradigm.

The second watermark works as a supporting media to share data between the embedding and exhibition stages. This watermark contains crucial information to reveal the first watermark and thus determine video ownership. Any error of the extracted data of the watermark may lead to mistakes for ownership protection. For this reason, the second watermark is embedded by using a very robust method based on quantization index modulation (QIM) under dither modulation in the DCT domain and by using some spatiotemporal HVS criteria. Several tests were performed to confirm the proposed scheme's performance regarding robustness, imperceptibility, and computational cost. The tests to measure watermark imperceptibility include several objective metrics designed for still images and videos, i.e., temporal distortion, are also considered. The obtained results confirm that an observer does not readily perceive the embedded watermark. Experimental results regarding robustness show that the proposed method is robust against signal processing attacks and video-based operations such as transcoding and temporal desynchronization. The proposal's practicality was also proven since it was tested in videos with different visual features and spatial resolutions. Finally, when compared to state-of-the-art techniques with a similar purpose, our proposed method is highly competitive, as it can more quickly process a video frame.

1.1. Contributions

The overall contribution of this proposal is to design a robust and imperceptible video watermarking technique that is performed in the base-band domain and is not computationally expensive. To meet this objective, we present two novel techniques within the field of study. The first technique allows decomposing a grayscale image into a set of binary images, called temporal codes, by using a spatiotemporal masking model [18]. A contribution of this work is that it adapts the method in [18] to video watermarking, which was not its original purpose. The use of temporal codes has two advantages. As we will explain later, the temporal codes are created by using a random masking function that provides security and resilience to the proposed scheme against intracollusion attacks [19]. Because the same watermark is not inserted in all video frames, it is more difficult for an attacker to determine a watermark signal's presence within the video. A second advantage is that the use of temporal codes allows us to obtain robustness against temporal desynchronization attacks. However, the method in [18] also presents some drawbacks. The temporal codes were originally created for synthetic videos; thus, the imperceptibility in practical scenarios is not guaranteed, and the original method is not robust against intentional or unintentional distortions. To address these issues, we use the second novel technique: the *imperceptible–visible* paradigm. Chuang [20] initially designed this method for still images, but due to its computational cost, it has not received enough attention for its use in video watermarking. The *imperceptible–visible* method allows us to determine the most suitable location of the watermark from the visual characteristics of the host video frame and adjust its strength with some HVS criteria. This makes it possible to obtain robustness against common signal processing tasks and video-based operations, enabling its application in practical scenarios. Another contribution of this proposal is the reduction in the high computational costs involved in the *imperceptible–visible* paradigm, thus allowing its use in the field of video watermarking. We present a method where a frame-by-frame process does not determine the watermark location. Instead, this method considers the video scene as the basic processing unit, which reduces the computational complexity considerably. To the best of our knowledge, there is no previously published work that combines temporal codes and the *imperceptible–visible* approach in the field of video watermarking.

1.2. Organization

This paper is organized as follows. In Section 2, we introduce the method to generate temporal codes from the first watermark. Section 3 presents the process to determine the location for both watermarks. We detail the proposed watermarking method in Section 4. Then, we present the experimental results in Section 5. Finally, we offer our conclusions in Section 6.

2. Related work

Recently, a work aiming at hiding a *secret message* along with a digital video by using spatial and temporal visual masking has been proposed in the scientific literature [18]. In general terms, the algorithm introduced in [18] initially considers the *secret message* as an input image with a color depth of 8 bits (grayscale image). The algorithm design premise is that the input image should be invisible in the same way for a single frame and the temporal addition mechanism of the HVS. We apply a self-masking model to meet this premise. First, to improve the effect of hiding information, or to reach the *masking threshold*, the method creates an image with a reduction of contrast I_c from the input image I, as follows:

$$I_c(x,y) = \alpha \cdot I(x,y) + \frac{1}{2} - \frac{\alpha}{2}, \tag{1}$$

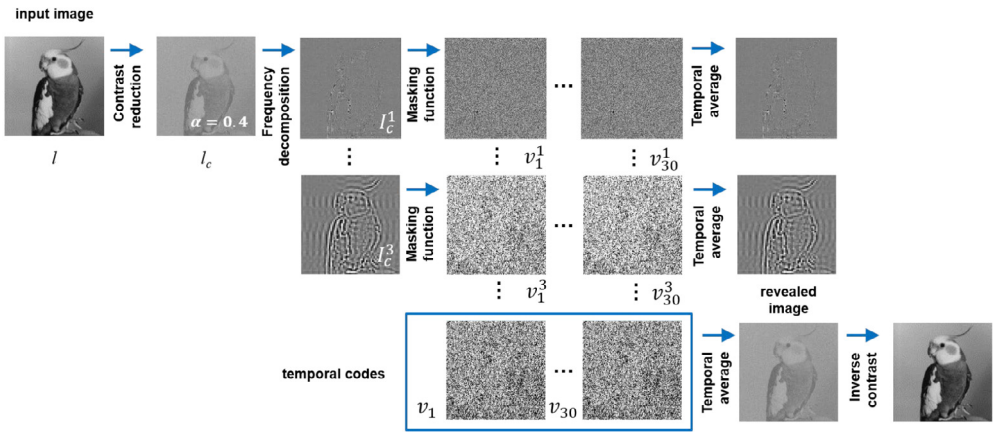

Fig. 1. Overview of the algorithm [18]. An input image I (visible watermark) is decomposed into $n = 30$ temporal codes by employing $k = 3$ frequency bands.

where α is the contrast reduction factor and takes a value between 0 and 1. Then, the low contrast image I_c is decomposed into k spatial frequency bands by applying the Laplacian pyramid method proposed in [21]. For each pixel value of the contrast-reduced image $I_c^l(x, y)$ in each l frequency band, Eq. (2) is fulfilled. This denotes that the n temporal samples created with a selected *masking function* can be integrated over time and give the corresponding frequency band l:

$$I_c^l(x, y) = \frac{1}{n}\sum_{i=1}^{n} v_i^l(x, y) \quad , \tag{2}$$

$$v_i^l(x, y) = f(t_i + \zeta_l(x, y)) \quad , \tag{3}$$

where v_i^l is a given frequency band l of frame v_i at time t_i of the video. The term $f(t+\zeta)$ refers to the *masking function* that operates by modifying each pixel (x, y) in each band l with a different phase shift value ζ. The authors in [18] recommend the adoption of three methods [22,23]: (a) random masking function, (b) sinusoidal composite wave, and (c) temporal ditter masking function. Finally, once the k frequency bands are created for each frame, they can be summed to obtain the final temporal code v_i at time t_i that holds information invisible to the naked eye, as follows:

$$v_i = \sum_{l=1}^{k} v_i^l. \tag{4}$$

When necessary, temporal averaging reveals the *secret message*. For a random masking function or sinusoidal composite wave, pixelwise mathematical averaging displays the input image. For the temporal ditter masking function, the authors [18] suggested disclosing hidden information via prolonged exposure by using a conventional camera lens. The experimental results presented in [18] show that the proposed algorithm's camouflage capacity is excellent, as is the lossless reconstruction when recovering the embedded image. However, several drawbacks may limit the application of the algorithm proposed in [18] by considering conventional digital watermarking requirements. First, since the videos generated by the algorithm in [18] are only synthetic, their use is limited in practical scenarios. Second, there is no evaluation of the algorithm against intentional or unintentional distortions; in essence, the algorithm's robustness is not proven. Based on this analysis, in this proposal, we employ the algorithm proposed in [18] but improve some of its limitations. We utilize the random masking function to create n temporal codes from a visible watermark signal, which contains the ownership information and is represented by a grayscale image. By randomly varying each pixel value of each frequency band l with uniformly distributed samples, we create the random masking function. It is important to note that Eq. (2) holds an error that has an indirectly proportional relationship with the number of temporal samples. In this way, a suitable number of temporal samples n must be experimentally determined for each application.

By adjusting the original method proposed in [18] with our proposed strategy, we obtain robustness against common signal processing tasks and video-based operations. This strategy also enables the application of the original method in practical applications with both conventional and synthetic videos. Fig. 1 graphically illustrates how to convert visible watermark I to its contrast reduction version I_c and the n temporal codes created using a random masking function. We can implement the inverse process of revealing the original watermark by applying a temporary average of the temporal codes with a pixelwise mathematical operation and a contrast reduction inverse operation.

3. Watermark location

In the proposed scheme, we embed two watermark signals in each video frame. First, a grayscale watermark image with ownership information is processed to generate n temporal codes according to the previous section's algorithm [18]. Once the temporal codes are created, they are embedded redundantly along with the video sequence with an *imperceptible–visible* watermarking approach. Under normal viewing conditions, the naked human eye cannot notice the visual quality distortion generated by the watermark embedding process. According to this model, the watermark must be easily perceived by performing common image-related functions [24]. In our proposal, the imperceptible–visible watermark is revealed by applying a binarization function that requires some crucial parameters embedded invisibly as a second watermark.

The selection of the locations of the two watermark signals is crucial since it considerably influences a robust, invisible, and less time-consuming proposal. This section describes in detail the considerations that must be addressed to select the best location for each watermark. Fig. 2 shows the overall strategy to choose the location of the two watermark signals.

3.1. imperceptible–visible watermark

In the *imperceptible–visible* approach, the watermark location is dependent on the visual features of the host frame. However, processing each frame of a video sequence to obtain the watermark location is computationally expensive and inefficient. To avoid this, the method to determine the most suitable location of the imperceptible–visible watermark is not a frame-by-frame process; rather, it is a procedure that uses the video scene as the basic processing unit and not the video frame. In other words, we determine the watermark location by performing an exhaustive search of the visual features of only one

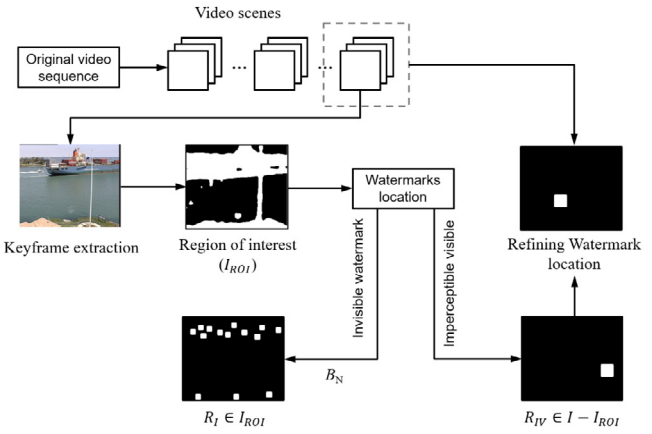

Fig. 2. The overall strategy to select the location of the watermarks.

frame (called keyframe) per video scene. This process includes five tasks: (a) splitting the video into scenes, (b) extracting the keyframe, (c) determining the region of interest, (d) selecting the most suitable watermark location, and (e) refining the position of the watermark along with the video sequence.

3.1.1. Video scene detection

Unlike still images, a video sequence incorporates temporal redundancy, or a lot of visual similarities between successive frames. Scene detection is a process that clusters the video into groups of frames with related visual features. The *imperceptible–visible* watermark location depends on each frame's visual features so that performing a scene detection can highly reduce the involved computational cost, as the watermark location of clustered frames is very similar. The process to split the video into scenes is based on a simple and efficient method presented in [25]. The first step is to replace each I-frame in the context of a group of images (GOP) of a video with its smaller version, called DC-frame [26], which is the average of intensities of every 8×8 block. This process reduces 64 times the amount of information to be processed. Despite having a low resolution, it only decreases 7% of a human's ability to appreciate the frame's visual details [27]. A color descriptor C is then computed from each DC-frame's chrominance component, together with masking and the average optical density operations. The reader is referred to [25] for a full report of the process to compute the mentioned color descriptor. The consecutive frame descriptors are then compared by using the L2 norm according to (5),

$$D[C_t, C_{t-1}] = \left(\sum_{r=1}^{u} \left| \left(C_t^r - C_{t-1}^r \right)^2 \right| \right)^{1/2}, \tag{5}$$

where C^r is the rth element of C, and C_t and C_{t-1} correspond to the descriptors C in time t and $t-1$, respectively. u denotes the length of C. The value of D remains near zero, while the consecutive frames have similar visual features. An increased value of D means significant visual changes, which correspond to a video scene change detection.

3.1.2. Key-frame extraction

Once a video scene is detected, the video frame in the middle of such a scene usually enough represents its visual content to be selected as the keyframe. It has been determined experimentally [25] that, on average, a video with a length of 2 min and a frame rate of 30 frames per second (*fps*) can be well-represented with a set of 20 keyframes, which is less than the 1% of the original video data. Considering this large computational cost reduction, in our proposal we only process each scene's keyframe to determine the best watermark location. Then, the keyframe's watermark location is used for all the frames of a scene since, as we explained before, the clustered frames of a video scene are visually similar. However, in videos where objects have much movement, the same watermark location would be inadequate for all frames, especially in those far from the middle of a scene. We overcome this problem by using a motion estimation technique that is explained later.

3.1.3. Region of interest

The region of interest is defined as the area of a frame that first attracts an observer's attention. The process to detect the region of interest is performed by employing a method to identify the saliency region in a frame I that is based on the image signature descriptor (*IS*) [28]. The *IS* is employed to build a frame $\overline{I}$, where the foreground information of I is estimated by performing the sign function to the 2D-DCT transform and then applying the inverse 2D-DCT operation (6):

$$\overline{I} = \mathrm{IDCT}\left[sign\left(\mathrm{DCT}\left(I \right) \right) \right]. \tag{6}$$

Each component of $\overline{I}$, denoted as $\overline{I}_i$, is convolved with a Gaussian filter K_σ of standard deviation σ to get a saliency map S_m.

$$S_m = K_\sigma * \sum_i \left(\overline{I}_i \circ \overline{I}_i \right). \tag{7}$$

In (7), the Hadamard product operator is represented with the symbol $\circ$, and the character $*$ denotes convolution. The standard deviation σ of the Gaussian kernel regulates the sensitivity of the blurring effect of the saliency map S_m. Finally, a binary image I_{ROI} is computed to isolate the foreground information of the saliency map S_m, by using a threshold T, as follows:

$$I_{ROI}(x,y) = \begin{cases} 1, & S_m(x,y) \geq T \\ 0, & S_m(x,y) < T \end{cases}, \tag{8}$$

where $x = 1, \ldots, M$, and $y = 1, \ldots, N$, and $M \times N$ denote the dimensions of I.

3.1.4. Watermark location

In this stage, we determine the most suitable location for the imperceptible–visible watermark, denoted as R_{IV}. The region R_{IV} must satisfy $R_{IV} \in (I - I_{ROI})$. The imperceptible–visible watermark is embedded within an area that is out of the region of interest for an observer. The above strategy improves imperceptibility, thus reducing the possibility of being perceived by the naked eye. The luminance component of the keyframe is evaluated to obtain the region with the lowest variance from all candidate regions R_{IV}^* by using:

$$R_{IV} = \arg\min_{R_{IV}^*} \left(\frac{1}{A \times B - 1} \sum_{(i,j) \in R_{IV}^*} \left(\rho_{i,j} - \mu_{R_{IV}^*} \right)^2 \right), \tag{9}$$

where $A \times B$ denotes the dimensions of the watermark, $\rho_{i,j}$ corresponds to the (i,j)th luminance pixel value, and the mean value of a candidate region R_{IV}^* is denoted as $\mu_{R_{IV}^*}$. Note that the computed region R_{IV} is a pixel-based estimation approach that seeks the frame's smoothest region in terms of variance.

3.1.5. Refining the watermark location

In a motionless video, the estimated watermark location for the keyframe is suitable for all frames of a video scene. However, some video scenes have movement of objects along video sequences, and in that case, the same watermark location could not be suitable for all frames. The proposed scheme uses a block matching method based on motion estimation to face the above issue. It is assumed that the objects that present motion within a frame can be traced to a corresponding block on a subsequent frame. Thus, the watermark location is considered a background block that can present motion between successive frames slightly. The original watermark location is refined by

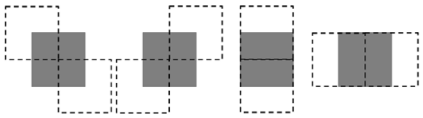

Fig. 3. The original watermark location (represented with a gray block). The eight neighborhood blocks (dotted line), used to compare the variance value.

comparing the original location's variance against eight neighborhood blocks (Fig. 3), with an offset of one pixel. If the variance of some of the eight compared blocks presents less variance than the original one, then the location is updated. Since the watermark location is computed for the keyframe located in the middle of a scene, frames must be divided into two sets to update the watermark location. The frames from the center to the end of a scene are compared with their previous frame. Similarly, those from the beginning to the middle of a scene are compared with their successive frame.

Fig. 4 shows the obtained result by applying the watermark location process to two video sequences with spatial dimensions of $M = 352$ and $N = 288$. From left to right, Fig. 4 shows the keyframe of the first scene, the keyframe's saliency map, and the keyframe's watermark location (denoted with a white block, A=B=24). Next, Fig. 4 illustrates the first frame of the same scene and the updated watermark location. The upper row of Fig. 4 shows the *news* video sequence with a static background, where the original watermark location remains unchanged throughout the video scene. The lower row of Fig. 4 corresponds to the *coastguard* video sequence with high motion. In this case, we can appreciate that the location was updated horizontally by following the smoothest region on the frame.

3.2. Invisible watermark

The watermark embedded with an invisible approach contains crucial information to determine video ownership. In this case, any variation of the extracted watermark may lead to errors of ownership protection. For this reason, this watermark is embedded by considering a very robust method performed in the DCT domain and using some spatiotemporal HVS criteria, together with the QIM method [29]. Importantly, we determine the location of the invisible watermark so that it belongs to the region of interest. An aggressive attack usually avoids causing significant visual quality degradation of the area of interest; otherwise, the video's value could be degraded. The invisible watermark's location is determined using an owner's secret key to select B_{N} blocks of 8×8 pixels randomly. The area represented by the B_{N} selected blocks is denoted as R_{I} and, as we have mentioned, meets the condition $R_I \in I_{ROI}$. This condition ensures that the visible and invisible watermark areas do not overlap.

4. Watermark embedding stage

This section explains in detail the process to embed both watermark signals in the video sequence. The watermark with ownership information is a grayscale image processed by the method described in Section 2. The process's output is the creation of n temporal codes, embedded along with the video sequence. This approach has several advantages, such as obtaining a resilient scheme against collusion attacks since the embedded watermark is different for the same scene's frames. Another advantage is that a suitable value of n allows a tolerance of the scheme against temporary desynchronization attacks, which will be addressed later. The first watermark, or the temporal code, is denoted as W_{IV}. The second watermark W_{I} is a binary sequence used to support media to share data between the embedding and exhibition stages. Both watermarks are embedded so that they are imperceptible. Fig. 5 shows a brief representation of the proposed watermarking embedding process. In Fig. 5, the dotted line denotes tasks that are performed at the frame level.

4.1. imperceptible–visible watermark

The process of embedding the imperceptible–visible watermark (W_{IV}) employs the Just Noticeable Distortion (JND) criteria. Notably, the JND criteria determine the most appropriate watermark strength value to guarantee the watermark's imperceptibility. The W_{IV} pattern is embedded in the host region R_{IV}, as follows:

$$R_{IV}^{w}(i,j) = \begin{cases} \max\left(0, \mu_{IV} - \left[\frac{T_W}{2}\right]\right), & W_{IV}(i,j) = 0 \\ \max\left(\mu_{IV} + \left[\frac{T_W}{2}\right], 255\right), & W_{IV}(i,j) = 1 \end{cases}, \tag{10}$$

where R_{IV}^{w} is the watermarked version of the host region R_{IV}, T_W is the JND criteria used to adjust the watermark strength, the term μ_{IV} is the intensity mean value of R_{IV}, and $[a]$ is an operation that calculates the nearest integer value of a.

The strength to embed the imperceptible–visible watermark highly depends on the mean intensity value μ_{IV} of the host region R_{IV} to keep imperceptibility. At this point, two considerations are taken into account. First, the HVS has more sensitivity in the middle-intensity level than in the boundaries. Second, the JND threshold represents the difference between the foreground's intensity level and the intensity of the background that the HVS cannot perceive. Considering the above, we determine the most appropriate JND value using the model introduced in [30]. Fig. 6 shows T_W's values regarding the host region's background intensity value, denoted as p. In this way, a background intensity value of 64 takes the lowest value of the JND threshold.

4.2. Invisible watermark

The process of embedding the invisible watermark contains four steps. First, the invisible watermark W_{I} is a binary representation of three data obtained from the imperceptible–visible watermark embedding process. Those data are the mean value μ_{IV} and the upper-left corner coordinates of R_{IV} (denoted as r_1 and c_1). The mean value μ_{IV} is in a range of [0,255], and the values of c_1 and r_1 are in a range of [0,2048] (the Full High Definition resolution is the highest spatial resolution considered in this proposal). Fig. 7 shows how the invisible watermark W_{I} is composed. Second, we determine the watermark location for the invisible watermark R_I by using an owner's secret key to randomly choose B_{N} blocks of 8×8 pixels from the luminance component that are within the limits of the region of interest ($R_I \in I_{ROI}$). One watermark bit is embedded in each block so that region R_{I} is composed of $B_{\mathrm{N}} = 30$. Third, we compute the 2D-DCT transform of each B_{N} block of 8×8 pixels that belongs to R_{I}. Finally, we experimentally determine [29] that the second alternating current ($AC_{1,2}$) coefficient of each 2D-DCT block is the most resilient coefficient against several aggressive operations, such as the quantization process at very low bit rates. Considering the above, each watermark bit is embedded into the $AC_{1,2}$ coefficient of each 2D-DCT block belonging to R_{I} by using the QIM algorithm [31], as follows:

$$AC_{1,2}^{w} = \begin{cases} \mathrm{sign}\left(AC_{1,2}\right) \times \left\lfloor \frac{|AC_{1,2}|}{2\varphi_{i,j}} \right\rfloor \times 2\varphi_{i,j}, & W_I^k = 0 \\ \mathrm{sign}\left(AC_{1,2}\right) \times \left(\left\lfloor \frac{|A_{1,2}|}{2\varphi_{i,j}} \right\rfloor \times 2\varphi_{i,j} \right) + \varphi_{i,j}, & W_I^k = 1 \end{cases}, \tag{11}$$

where $AC_{1,2}$ and $AC_{1,2}^{w}$ are the original and watermarked DCT coefficients, respectively. W_I^k is the kth bit of the watermark W_{I}, and $\varphi_{i,j}$ is the QIM quantifier that is computed by using a static QIM quantifier Q and a saliency-modulated JND [32]:

$$\varphi_{i,j} = S(\mathrm{JND}(t,n,i,j)) \times Q. \tag{12}$$

The most suitable value for the static QIM quantifier Q is set regarding the trade-off between imperceptibility and robustness. The term $S(\mathrm{JND}(t,n,i,j))$ is a saliency modulated JND threshold for the frame

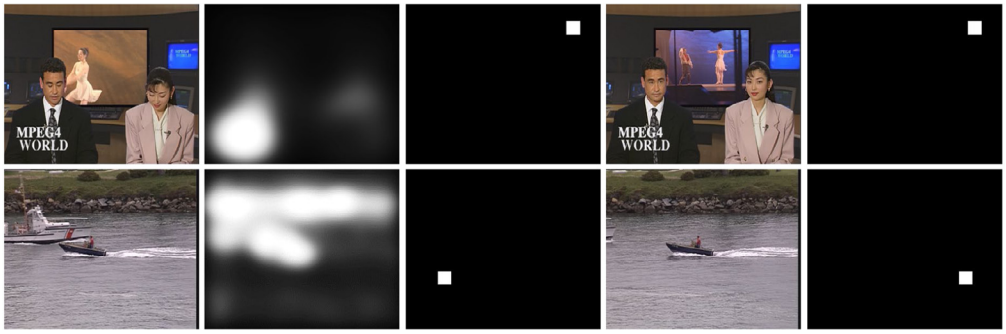

Fig. 4. The proposed watermark location process applied to the *news* (upper row) and *coastguard* (low row) video sequences. From left to right: the first scene's keyframe, its saliency map, the watermark location for the keyframe (white block), the scene's first frame, and the updated watermark location (white block).

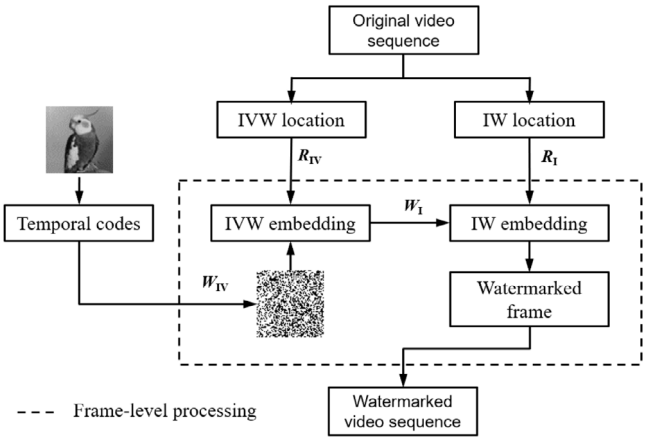

Fig. 5. The proposed watermark embedding process.

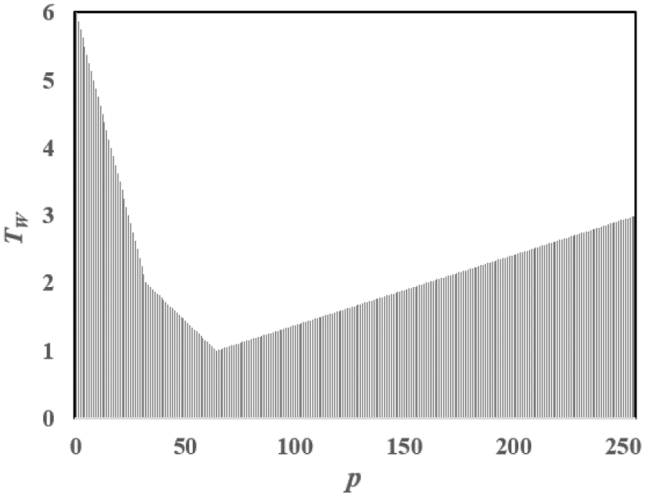

Fig. 6. The JND threshold (T_W) for each background intensity value p.

at time t, in the block n, and the (i, j) 2D-DCT coefficient, which is defined as follows:

$$S\,(\mathrm{JND}\,(t,n,i,j)) = T(t,n,i,j) \times \rho_{CONT}(t,n,i,j) \times \rho_{LUM}(t,n) \times \Psi^M(t,n), \tag{13}$$

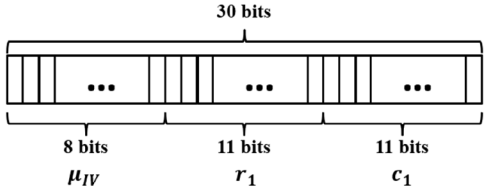

Fig. 7. The binary representation of the invisible watermark W_I.

where $T(t,n,i,j)$ is a base JND value that considers the spatiotemporal contrast sensitivity function (CSF), the gray intensities of the frame, and the oblique and spatial summation factors [33]. The terms ρ_{CONT} and ρ_{LUM} denote contrast and luminance marking operations, respectively. Finally, an empirical linear function Ψ^M is used to adjust the JND value regarding the saliency area of the frame [32]. Since the invisible watermark is located within the region of interest, the saliency modulated JND aims to adapt the QIM quantifier to obtain a highly robust scheme with minimal distortion of the visual quality.

5. Watermark exhibition stage

The first step of the exhibition stage, which reveals the embedded imperceptible–visible watermark pattern, is to obtain the crucial parameters by performing the invisible watermark extraction process. The content owner's secret key is used to locate the B_N blocks where the watermark was previously embedded. Then, the extracted binary watermark $\hat{W}_I$ is obtained by applying the QIM extraction process [31], as follows:

$$\hat{W}_I(k) = \begin{cases} 0, & \text{if round}\left(AC^w_{1,2}/\varphi_{i,j}\right) = \text{even} \\ 1, & \text{if round}\left(AC^w_{1,2}/\varphi_{i,j}\right) = \text{odd} \end{cases}, \tag{14}$$

where $\hat{W}_I(k)$ is the kth bit of the extracted watermark, $AC^w_{1,2}$ is the watermarked DCT coefficient, and $\varphi_{i,j}$ is the QIM quantifier previously calculated in the embedding process and used as a secret key at this stage. Once $\hat{W}_I$ is extracted, the values of $\hat{\mu}_{IV}$ and $(\hat{r}_1, \hat{c}_1)$ are retrieved, which represent the mean value and the upper-left corner coordinates of R^w_{IV}, respectively. The extracted mean value $\hat{\mu}_{IV}$ is used to build a binary image from the luminance component $\hat{I}_Y$ of each watermarked frame by using:

$$\hat{I}_W = \begin{cases} 1, & \text{if } \hat{I}_Y \geq \hat{\mu}_{IV} \\ 0, & \text{otherwise} \end{cases}, \tag{15}$$

where $\hat{I}_W$ is the watermarked binary image with the revealed watermark, and the temporal code is visible to the naked eye at the $(\hat{r}_1, \hat{c}_1)$ coordinates. Then, with the values of $(\hat{r}_1, \hat{c}_1)$ and the dimensions

Fig. 8. A graphical example of the exhibition stage. The upper row shows the watermarked frames, at times *t=1, t=120, t=240, and t=300*, of the *akiyo* video sequence. The middle row shows the binary image representation $\hat{I}_W$ for the same frames. Finally, the lower row shows the binary images $\hat{I}_W$, but the region R_{IV}^w is changed by B_{IV}^w. In the middle and lower rows, the regions R_{IV}^w and B_{IV}^w are zoomed and displayed at the bottom-right corner for better appreciation.

W_{IV}, the region R_{IV}^w is isolated to perform a temporal adding of such a region:

$$B_{IV}^w = \sum_{t=1}^{n} R_{IV}^w(t) \quad , \tag{16}$$

where B_{IV}^w is a grayscale image that contains the sum of all the binary watermarked regions R_{IV}^w from $t = 1$ to n. Note that n is the number of temporal codes generated from the original grayscale watermark. In the process, the value of t is reset when t=n or when the end of a scene is reached. To get a better understanding of the exhibition stage, Fig. 8 shows an example of the watermarked version of the *akiyo* video sequence. The upper row shows the watermarked frames at times $t = 1$, $t = 120$, $t = 240$, and $t = 300$ from left to right, respectively. The middle row of Fig. 8 consists of the binary images $\hat{I}_W$ at the same values of t. In this experiment, the *akiyo* video sequence has spatial dimensions of 704×576 pixels and consists of 10 s of video at 30 *fps*. Considering this, the original watermark (A=B=64) is processed to build $n = 300$ temporal codes. For demonstrative purposes, the region R_{IV}^w is zoomed and displayed at the bottom-right corner. Finally, the lower row of Fig. 8 shows the binary images $\hat{I}_W$ at the same values of t but with the difference that the region R_{IV}^w switches to the values of B_{IV}^w at times $t = 1$, $t = 120$, $t = 240$, and $t = 300$, respectively, by using (16). In the middle row, the region B_{IV}^w is also zoomed and displayed at the bottom-right corner.

6. Experimental results

In this section, we perform several experiments to evaluate the effectiveness and performance of the proposed method. To conduct our experiments, we create a database of 20 videos, codified under the MPEG-4 Part 2 compression standard by using the ASP Profile. The dataset is composed of 8 videos with a spatial resolution of 352×288 pixels (CIF Format) at 30 *fps*, 8 videos with a spatial resolution of 704×576 pixels (4CIF Format) at 24 *fps*, and 4 videos with a spatial resolution of 1920×1080 pixels (HD Format) at 30 *fps*. These videos were selected by considering the amount of movement within the video sequence and their lighting, texture, and color conditions. This will prove the application of the method in practical scenarios. A sample of the videos used in this paper can be observed in Fig. 9, and the rest of the videos are shown as part of the experiments.

The watermarks used in our experiments are shown in Fig. 10; in the tests, the watermark size is adjusted according to the host video dimensions.

6.1. Watermark strength

The proposed method embeds two watermarks in each video frame, and the strength of those watermarks has to be determined to obtain a robust watermark while maintaining imperceptibility. According to the host frame's visual features, the strength of the imperceptible–visible watermark (WIV) is determined by using JND criteria. In the case of the invisible watermark (W_{I}), the quantifier Q is empirically chosen by considering the trade-off between imperceptibility and robustness. The value of Q is as high as possible, as long as the watermark remains imperceptible to the naked eye. To determine the optimal value of Q, we perform the invisible watermark embedding process described in Section 4.2 with Q values ranging from 10 to 20. Then, we measure the effect of the visual quality degradation by using the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) [34] metrics. The robustness is measured by using the bit error rate (BER) under the same conditions. Table 1 shows the obtained values for each metric. Values in Table 1 correspond to the average data of all tested videos. Table 1 shows the optimal value of Q as 16 (underlined data line), and the PSNR and SSIM values of 51.07 and 0.9914, respectively, confirm that the watermark is not readily perceptible. Note that a value of SSIM equal to 1 indicates that two frames are identical. Under these conditions, the BER is equal to 0. The previous Q value also has this value, which suggests that the scheme tolerates more noise generated from attacks, and the watermark could be entirely recovered.

6.2. Watermark imperceptibility

This section measures the visual quality distortion generated on the video sequence due to performing the watermark embedding process of both the imperceptible–visible watermark W_{IV} and the invisible watermark W_{I}. The metrics employed are the PSNR, the SSIM, the video

Fig. 9. A sample of the videos used to carry out our experiments.

Fig. 10. The grayscale watermarks used in our experiments.

Table 1
Quality metrics results to determine the QIM quantifier Q.

Q	PSNR (dB)	SSIM	BER
10	54.62	0.9954	1.03 E−08
12	53.09	0.9943	3.51 E−09
14	51.82	0.9921	0
16	51.07	0.9914	0
18	49.23	0.9886	0
20	48.72	0.9864	0

Table 2
Quality degradation after performing two watermark embedding processes.

PSNR	SSIM	VQM	**stVSSIM**
51.07	0.9914	0.32	0.9630

quality model (VQM) [35], and the spatiotemporal video SSIM (stVSSIM) [36]. The VQM is a DCT-based quality metric oriented to video, as it considers the spatiotemporal CSF. The value of VQM rises as the quality of video decreases; importantly, a value of 0 means a losslessly compressed video. On the other hand, the stVSSIM metric considers the motion information between frames, enabling us to measure the temporal video distortion. stVSSIM uses the same scale as spatial SSIM, where a value of 1 indicates identical videos. After applying the cited metrics, the obtained results are shown in Table 2, representing the average of all the video sequences tested in our experiments. According to Table 2, it is possible to determine that an observer cannot, with the naked eye, perceive the differences between the original and watermarked videos at the frame level while the video is in motion.

Fig. 11 shows an example of the proposed method by applying it to videos with different visual features. From left to right, Fig. 11 shows the last frame of the video sequence, its watermarked version, the binary image representation $\hat{I}_W$, and the image $\hat{I}_W$ with the region B^w_{IV} computed by adding the R^w_{IV} areas temporally. The upper row of Fig. 11 shows the *Stefan* video sequence, which is a sequence of 10 s at 24 *fps* with a spatial resolution of 704 × 576 pixels. This video is textured and has content with bright colors. The proposed method works as expected since the R_{IV} region is the area with less variance from the frame, and the naked eye cannot easily perceive it (PSRN=50.45, SSIM=0.9903, VQM=0.39, and stSSIM=0.9549). At the end of the process, it is possible to clearly distinguish the watermark (A=B=64), enabling the possibility of determining the media's ownership. On the other hand, the lowest row of Fig. 11 shows the *Miss America* video sequence. The video presents different characteristics since the background is mainly dark and not textured. The watermark remains the same size (A=B=64) and appears to be larger than in the previous experiment because the spatial resolution of the *Miss America* video is 352 × 288 pixels. Again, the proposed method finds the most suitable location for the imperceptible–visible watermark by looking for the region with less variance, regardless of whether such a region is dark or bright. The quality measurement values confirm that the watermark is invisible to an external observer (PSRN=51.53, SSIM=0.9931, VQM=0.29, and stSSIM=0.9829).

6.3. Watermark robustness

A watermarked video can be affected in practical situations by performing intentional or unintentional tasks that can partially or completely remove the previously embedded watermarks. This section evaluates the robustness of the watermark against common signal processing tasks and video-based operations. For all simulated hostile operations, it is necessary to test the robustness of both watermarks. First, the BER value of the invisible watermark W_{I} is evaluated to determine if the crucial parameters can be properly extracted to reveal the second watermark. Second, since some attacks can alter the video frame's visual features, it is necessary to confirm that the imperceptible–visible watermark W_{IV} is visible after the exhibition stage.

Concerning signal processing tasks, we tested the scheme by simulating the addition of impulsive noise contamination and image enhancement via a sharpening operation. Fig. 12 shows the results of the robustness obtained against (a) impulsive noise contamination with a density ranging from 0 to 0.1 and (b) a sharpening effect with a window size varying from 1 to 7. Impulsive noise contamination is introduced when the video is transmitted over a noisy channel. It is manifested as a random variation of a pixel's intensity level and degrades the video's visual quality. From Fig. 12(a), we can appreciate that the proposed scheme is very robust against this attack. The obtained BER values remain lower than 2% when the noise contamination density is approximately 10%. On the other hand, sharpening is a common signal processing operation carried out intentionally. It involves applying a high pass filter to the video frame to bring out its features by increasing the contrast between bright and dark regions. Again, the proposed scheme is highly robust against this attack, as shown in Fig. 12(b), where a window size of 7 generates a BER value of only 3%. Fig. 13 shows the result of the exhibition stage of the imperceptible–visible watermark. The upper row of Fig. 13 presents the impulsive noise contamination with a density of 9% applied to the *highway* video sequence, which has a spatial resolution of 352 × 288 pixels at 30 *fps*. The lower row of Fig. 13 shows an example of the sharpening operation with a window size of 5 applied to the same video sequence. From left to right, Fig. 13 shows the last watermarked frame of the video (which includes the simulated attack), the binary version ($\hat{I}_W$) of the same frame, and the binary version ($\hat{I}_W$) with the computed region R^w_{IV}. As we can appreciate, in both cases, it is possible to see the revealed watermark at a glance, which indicates that the scheme is robust against these hostile operations.

Fig. 11. An example of the proposed method by applying it to videos with different visual features.

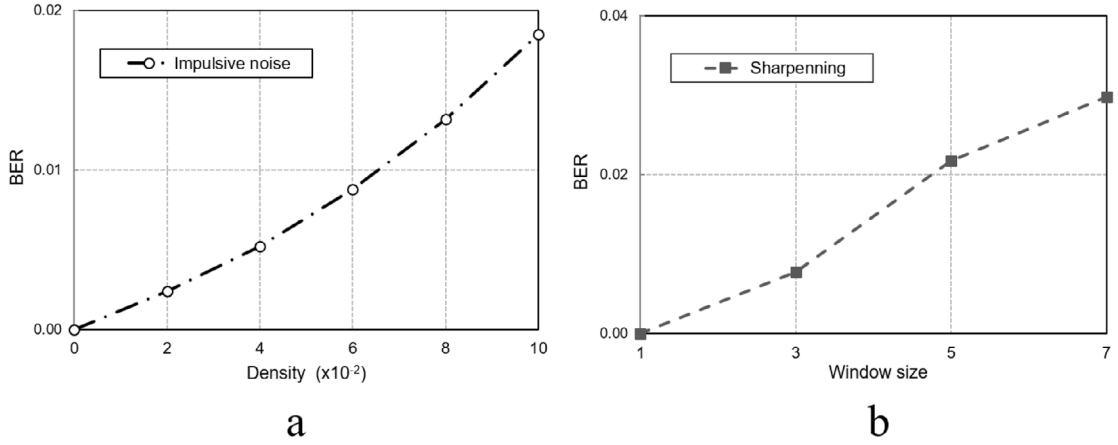

Fig. 12. The robustness performance of the proposed watermarking scheme against (a) impulsive noise contamination and (b) sharpening.

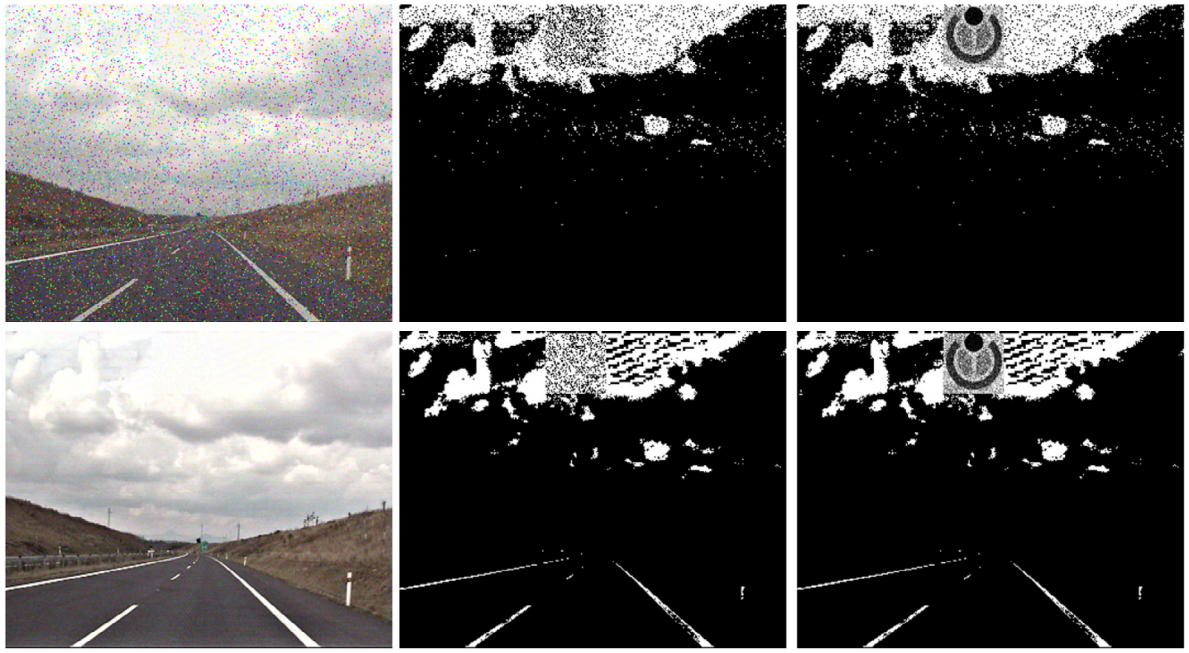

Fig. 13. Applying the impulsive noise contamination (upper) and the sharpening (lower) operations to the *highway* video sequence. From left to right: the watermarked frame, its binary version ($\hat{I}_W$), and the binary version with the computed region R_{IV}^{w}.

Table 3
The robustness performance of the proposed watermarking scheme against transcoding.

Compression standard								
H.264 AVC			MPEG-4 Part 2			MPEG-2		
1 Mbps	2 Mbps	4 Mbps	1 Mbps	2 Mbps	4 Mbps	1 Mbps	2 Mbps	4 Mbps
0.0257	0.0145	0.0088	0.0108	0.0084	0.0054	0.0121	0.0091	0.0075

Regarding video-based operations, two hostile tasks can compromise the embedded watermark: transcoding and temporal desynchronization. Transcoding is a very aggressive operation that usually occurs unintentionally when an end user tries to play a video on different devices. To accomplish this operation, it is necessary to convert the source video format by changing its resolution, bit rate, video codec, or frame rate to adapt the video sequence to the new device's resources. To test the proposed scheme against transcoding operations, first, the watermark embedding process is performed to obtain a video sequence with lossless compression. Next, this video is transcoded by using FFmpeg software [37] to convert it to three different compression standards (H.264 AVC, MPEG-4 Part 2, and MPEG-2) with three fixed bit rates (4 Mbps, 2 Mbps, and 1 Mbps) to measure the robustness of the watermark under different transcoding scenarios. The result is presented in Table 3, which contains the average results of all tested video sequences. From Table 3, it is possible to determine that, as we mentioned above, the invisible watermark applied algorithm is very robust against transcoding operations. Therefore, it is possible to extract the crucial parameters since the most significant BER value has an approximately 2% error. Fig. 14 shows an example of the transcoding simulation explained above to confirm how visible the imperceptible–visible watermark is after this process. Fig. 14 shows the *mother-daughter* video sequence codified under the H.264 AVC compression standard (the most aggressive transcoding) with bit rates of 4 Mbps, 2 Mbps, and 1 Mbps (from top to bottom). Only a zoomed region of the video is shown to better appreciate the codification's visual quality degradation. The second column presents the result of the watermark exhibition stage, with the computed region R_{IV}^{w} zoomed at the bottom-right corner. From Fig. 14, it can be verified that the watermark is visible after all transcoding operations. The quality of the revealed watermark decreases when the bit rate is lower; however, it is possible to appreciate the associated ownership information in all cases. Thus, it is possible to claim copyright.

Finally, temporal desynchronization operations refer to the alteration of the order or amount of video frames. These operations are usually performed intentionally and become especially relevant when the watermark information is embedded in two or more consecutive frames. There are three main types of temporal desynchronization operations: frame dropping, frame averaging, and frame swapping. Frame dropping is an attack that occurs when some frames are intentionally deleted or lost during streaming. Frame averaging is an operation that averages the pixels of two successive frames at a specific frequency of time. This operation does not change the order of frames but may severely affect the features. Frame swapping aims to alter the order of video frames to change the result of the watermark detection process.

As we mentioned earlier, because the random masking function is employed to generate n temporal codes, the input image is revealed by pixelwise mathematical averaging. Therefore, changing the order of the video frames does not create any effect since the commutative law is satisfied. In this way, the robustness of the proposed method against temporal desynchronization attacks is determined by measuring the watermark exhibition stage's tolerance against the loss of frames, as frame averaging can be considered the loss of one of the averaged frames. Fig. 15 shows an example of the effect caused by the exhibited watermark by the loss of frames. As can be observed, the watermark loses visual quality as the percentage of video frames increases and becomes imperceptible when half of the video frames have been lost. However, it has been experimentally determined that a higher drop of 20% of the frames produces a *jerky* effect on the video, thereby diminishing its commercial value [38].

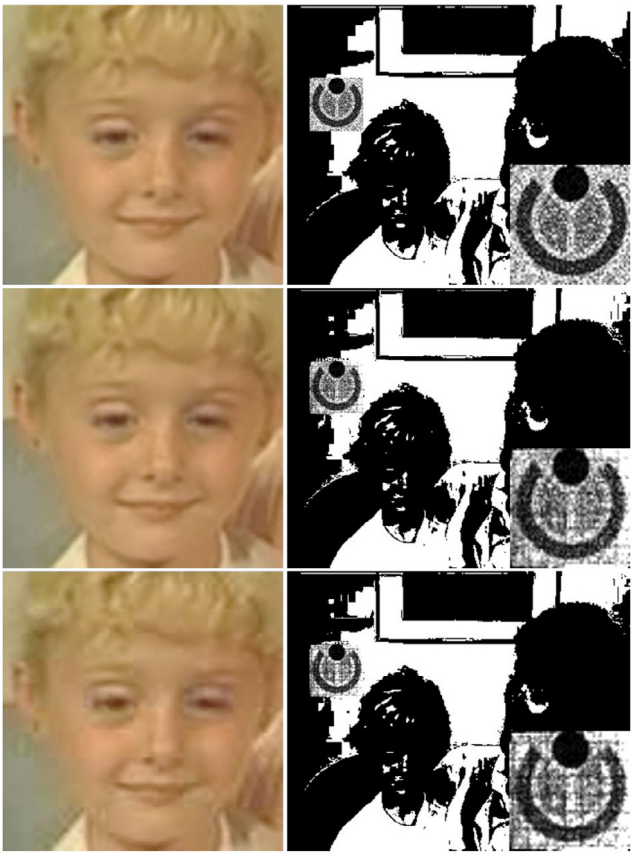

Fig. 14. An experiment to determine the robustness of the proposed scheme against transcoding operations. The first column shows a zoomed region of the *mother-daughter* video sequence under the H.264 AVC compression standard with bit rates of 4 Mbps, 2 Mbps, and 1 Mbps (from top to bottom). The second column presents the result of the watermark exhibition stage, with a zoomed version of the computed region R_{IV}^{w} at the bottom-right corner.

6.4. Performance comparison

To highlight the relevance of the proposed paper's contributions, in this section, we conduct a performance comparison among the results obtained by the proposed scheme and those obtained by the methods in [32,39–43]. These methods perform video watermarking under comparable conditions and use similar metrics to assess imperceptibility and robustness. Since it is not easy to find methods that exactly apply all the robustness experiments involved in this work, the comparison exercise has been divided into two parts. The first comparison is performed using two signal processing attacks (impulsive noise and contrast adjustment) and the frame dropping temporal desynchronization operation. Table 4 shows the results of the first comparison.

Regarding imperceptibility, all methods obtain a PSNR value higher than 41 dB, suggesting that an observer does not readily perceive the watermark with the naked eye. In this case, all PSNR values were obtained before the attacks were performed. The method in [41] obtains the highest PSNR value, at 60.95 dB. However, it is well-known that the PSNR metric cannot fully characterize HVS properties and generates low-accuracy estimations regarding visual quality. The SSIM metric is regarded as a more reliable indicator of image quality. All methods perform excellently since the values are very close to 1. The work in [40] does not report the SSIM metric. According to Table 4, the proposed method achieves the best performance against impulsive noise contamination operation, with a density of 2%. This is notable if we consider that the methods in [40,41] report higher BER values at

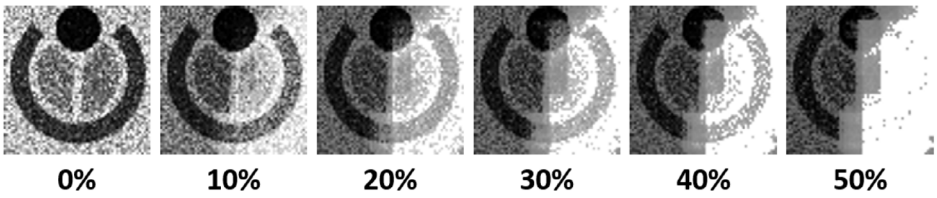

Fig. 15. An example of the effect caused in the exhibited watermark by the loss of frames (expressed as a percentage).

Table 4
State-of-the-art comparison ($^*d = 0.01$).

Algorithm	No attacks		BER		
	PSNR	SSIM	Impulsive noise (d=0.02)	Contrast adjustment	Frame Dropping (20%)
Proposed method	51.07	0.991	0.0024	0.0298	0.009
[39]	55.00	0.999	3.9068	4.9927	0.017
[40]	41.50	–	0.0303*	0.0217	0.126
[41]	60.95	1.000	0.0100*	–	0.082

Table 5
State-of-the-art comparison regarding transcoding operations.

Algorithm	No attacks		BER	
	PSNR	SSIM	MPEG-2	MPEG-4
Proposed method	51.07	0.991	0.0054	0.0075
[32]	56.00	0.970	0.0090	0.0090
[42]	58.00	–	–	0.0270
[43]	56.18	–	0.0060	0.2100

a lower level of density noise (1%). A similar performance is reached against the contrast adjustment attack. In this case, the [40] method obtains a slightly lower BER value than the proposed method, which indicates good performance. The process in [41] does not report the contrast adjustment attack. Concerning frame dropping, all proposals obtain a noticeable performance, and this is important if we consider that the proposed method embeds the watermarking throughout the temporal domain.

In the second part of the comparison, again, we look for methods that present similar imperceptibility metrics, but in this case, we also look for metrics that present robustness results against transcoding tasks. The techniques in [32,42], and [43] are compared with our method by measuring the reported BER value after transcoding the watermarked video to the MPEG-2 and MPEG-4 video compression standards. The results of the second performance comparison are presented in Table 5. From Table 5, we can appreciate that all methods show PSNR values higher than 50 dB. The proposal in [32] is the only one that reports the SSIM value, reflecting good performance regarding imperceptibility. It is important to note that none of the methods in Tables 4 and 5 apply a metric that considers the temporal distortion of the video as the proposed method, which uses the VQM and stVSSIM metrics to present imperceptible results.

Regarding MPEG-2 video compression, all methods report good performance since the BER value is lower than 1% in all cases. The proposal in [42] does not report this attack. However, in the case of MPEG-4 compression, the methods in [42,43] raise the BER values to 2% and 21%, respectively. The proposed method and the method in [32] still present BER values lower than 1% against this attack, but again, the proposed method achieves the best performance.

6.5. Time consumption analysis

As mentioned throughout this work, one of the main objectives of our study is to present a video watermarking method that is not computationally expensive. To measure the time consumed by the proposed approach, we carry out an experiment by performing the watermark embedding process on 3 video sequences, each of which is 10 s long, at a frame rate of 30 *fps*, with spatial resolutions of 352×288, 704×576, and 1920 × 1080 pixels. We carried out the experiment using MATLAB® 2019a, running on an Intel® Core™ i5-8250U CPU at 1.60 GB of RAM. The results are shown in Fig. 16. The gray bars denote the time consumed by processing the 300 frames for each spatial resolution without using the proposed strategy explained in Section 3 (i.e., by performing an exhaustive search to determine the watermark location as a frame-by-frame task). The results are 63.83 min for a video with a spatial resolution of 1920 × 1080 pixels, 28.43 min for a video with a spatial resolution of 704 × 576 pixels, and 16.93 min for a video with a spatial resolution of 352 × 288 pixels. We repeat the experiment but consider the proposed strategy that suggests the watermark location only for each scene's keyframe and then refines this position along the video. The results are represented by the black bars in Fig. 16. The obtained results are notably lower than those previously mentioned. This experiment found numerical values of 3.49 min, 1.49 min, and 0.97 min for videos with spatial resolutions of 1920×1080, 704×576, and 352 × 288 pixels, respectively. In this way, compared to a traditional approach that processes all video frames, we present a method that saves time consumption by approximately 94%.

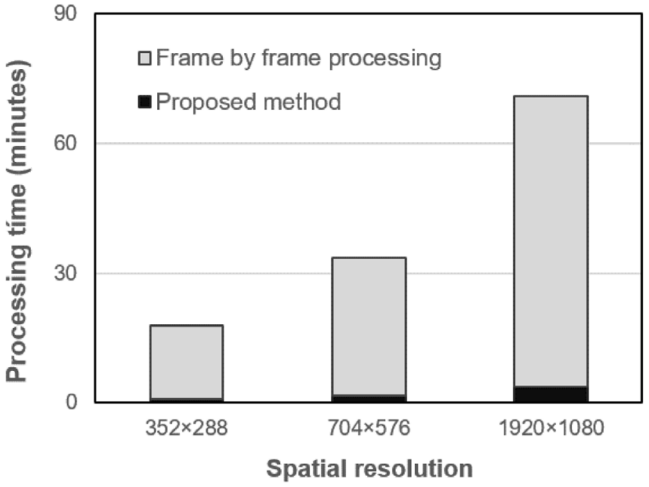

Fig. 16. Time consumption analysis.

By considering the three spatial resolutions tested in this work, our proposed scheme averages a time of 0.39 s for processing each frame of a video. To put this result in context, we compare it with other results reported in state-of-the-art techniques, as presented in

Table 6
Time consumption comparison.

Algorithm	Time consumption
Proposed method	0.39 s
[44]	10.1 s
[45]	0.35 s
[46]	3.26 s

Table 6. It is important to note that we avoid comparing real-time or hardware-based implementations since these proposals' goals are distinctly different from our own. In contrast, we compare the embedding time reported in general-purpose watermarking implementations [44–46] with the unique objective of highlighting the importance of our proposed method to save computational costs. The proposal presented in [44] is a watermarking technique based on the DWT domain. In [44], the authors reported that the time consumed by their watermark embedding process ranged from 10.1 to 30.5 s to process each frame. On the other hand, [45] presents a collusion-resistant video fingerprinting method that introduces an efficient scheme for the embedding process. The reported result in [45] is 5 min to process 852 VGA frames (i.e., 0.35 s per frame); however, this time does not consider higher spatial resolution tests that could cause the time to increase. Our proposed method takes 1.49 min to process 300 frames with a similar spatial resolution (704 × 576 pixels), which means an average of 0.29 s per frame. Finally, the method proposed in [46] presents a software implementation of a video watermarking scheme focused on videos encoded under the current H.265 AVC codec. In this case, the time to perform video compression increases by 3.2 to 11.4 s when we carry out the watermark embedding process. Table 6 shows that our proposed method is a fast and efficient solution. When compared to state-of-the-art techniques with a similar purpose, our proposed method more quickly and competitively processes each video frame.

7. Conclusions

In this paper, we propose a video watermarking method that addresses the problem of finding a method developed in the base-band domain that does not imply high computational costs, which is uncommon in the video watermarking research field. To address this problem, we rely on two innovative techniques in the field of study: temporal codes and the *imperceptible–visible* paradigm. One of the main contributions of this work is to adapt these methods to benefit from their advantages and implement some strategies to solve some of their disadvantages. To demonstrate the contribution of the proposed method, we carried out computer simulations regarding imperceptibility, robustness, and time consumption. Imperceptibility was measured by employing metrics designed for still images (PSNR and SSIM) and video sequences (VQM and stVSSIM). The obtained values for imperceptibility are a PSNR value higher than 50 dB, a SSIM value very close to 1, a stVSSIM value of 0.9630, and a VQM value of 0.32, which is very good considering that the VQM value rises rapidly as the quality of video decreases. These results confirm that the naked eye cannot perceive watermarks embedded in the test videos. Regarding robustness, the experimental results show a loss of only approximately 3% of the original watermark when the scheme is subjected to very aggressive signal processing operations such as impulsive noise contamination and image enhancement by sharpening. Video-based tasks were also simulated to confirm the robustness of the proposed scheme against transcoding by changing the video compression standard from lossless video to H.264 AVC, MPEG-4 Part 2, and MPEG-2 and changing the bit rate compression. The most aggressive transcoding operation was the H.264 AVC compression standard with a bit rate of 1 Mbps. In this scenario, the obtained BER value was less than 2%. Temporal desynchronization was also considered, and it has been experimentally determined that the watermark suffers considerable visual damage when the loss of the video frames reaches 50%. However, the loss of this number of frames is not a practical situation. Finally, the conducted experiments show that, compared to state-of-the-art techniques with a similar purpose, our proposed method is highly competitive, as it can more quickly process a video frame.

In this way, we have validated that the proposed design has an important implication in the overall performance of the solution. The obtained results confirm that the proposed scheme can be considered a suitable solution that could be applied in practical cases. Although many robust video watermarking techniques have been proposed, this research field still faces many challenges. Researchers should continue to focus on how to improve imperceptibility and robustness while also reducing computational costs. In this paper, we present an innovative solution that provides some ideas to solve this problem.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Authors thank the Instituto Tecnologico *y* de Estudios Superiores de Monterrey (ITESM), Mexico, the Consejo Nacional de Ciencia *y* Tecnologia (CONACYT), Mexico, as well as the Instituto Politecnico Nacional (IPN) by the support provided during the realization of this research.

References

[1] Q. Su, D. Liu, Z. Yuan, G. Wang, X. Zhang, B. Chen, T. Yao, New rapid and robust color image watermarking technique in spatial domain, IEEE Access 7 (2019) 30398–30409, http://dx.doi.org/10.1109/ACCESS.2019.2895062.
[2] Y. Liu, J. Zhao, A new video watermarking algorithm based on 1D DFT and radon transform, Signal Process. 90 (2010) 626–639, http://dx.doi.org/10.1016/j.sigpro.2009.08.001.
[3] J. You, J. Korhonen, A. Perkins, Spatial and temporal pooling of image quality metrics for perceptual video quality assessment on packet loss streams, in: International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2010, pp. 1002–1005..
[4] T. Dutta, H.P. Gupta, An efficient framework for compressed domain watermarking in P frames of high-efficiency video coding (HEVC), encoded video, ACM Trans. Multimed. Comput. 13 (2017) 1–24, http://dx.doi.org/10.1145/3002178.
[5] M. Fallahpour, S. Shirmohammadi, M. Semsarzadeh, J. Zhao, Tampering detection in compressed digital video using watermarking, IEEE Trans. Instrum. Meas. 63 (2014) 1057–1072, http://dx.doi.org/10.1109/tim.2014.2299371.
[6] S.J. Horng, M.E. Farfoura, P. Fan, X. Wang, T. Li, J.M. Guo, A low cost fragile watermarking scheme in H.264/AVC compressed domain, Multimed. Tools Appl. 72 (2013) 2469–2495, http://dx.doi.org/10.1007/s11042-013-1561-2.
[7] G. Yoo, H. Kim, Real-time video watermarking techniques robust against re-encoding, J. Real-Time Image Process. 13 (2015) 467–477, http://dx.doi.org/10.1007/s11554-015-0557-8.
[8] W. Wang, J. Zhao, Hiding depth information in compressed 2D image/video using reversible watermarking, Multimed. Tools Appl. 75 (2015) 4285–4303, http://dx.doi.org/10.1007/s11042-015-2475-y.
[9] W. Gao, G. Jiang, M. Yu, T. Luo, Lossless fragile watermarking algorithm in compressed domain for multiview video coding, Multimed. Tools Appl. 78 (2019) 9737–9762.
[10] M. Asikuzzaman, M.R. Pickering, An overview of digital video watermarking, IEEE Trans. Circuits Syst. Video Technol. 28 (2018) 2131–2153, http://dx.doi.org/10.1109/tcsvt.2017.2712162.
[11] F. Arab, S.M. Abdullah, S.Z.M. Hashim, A.A. Manaf, M. Zamani, A robust video watermarking technique for the tamper detection of surveillance systems, Multimed. Tools Appl. 75 (2015) 10855–10885, http://dx.doi.org/10.1007/s11042-015-2800-5.
[12] Z. Bahrami, F. Akhlaghian, A new robust video watermarking algorithm based on SURF features and block classification, Multimed. Tools Appl. 77 (2016) 327–345, http://dx.doi.org/10.1007/s11042-016-4226-0.
[13] A.A. Mohammed, N.A. Ali, Robust video watermarking scheme using high efficiency video coding attack, Multimed. Tools Appl. 77 (2017) 2791–2806.
[14] M. Asikuzzaman, M.J. Alam, A.J. Lambert, M.R. Pickering, Imperceptible and robust blind video watermarking using chrominance embedding: A set of approaches in the DT CWT domain, IEEE Trans. Inf. Forensics Secur. 9 (2014) 1502–1517, http://dx.doi.org/10.1109/tifs.2014.2338274.

[15] L. Tian, H. Dai, C. Li, A semi-fragile video watermarking algorithm based on chromatic residual DCT, Multimed. Tools Appl. 79 (2019) 1759–1779.
[16] I. Agilandeeswari, K. Ganesan, A robust color video watermarking scheme based on hybrid embedding techniques, Multimed. Tools Appl. 75 (2015) 8745–8780, http://dx.doi.org/10.1007/s11042-015-2789-9.
[17] W. Lu, W. Sun, H. Lu, Novel robust image watermarking based on subsampling and DWT, Multimed. Tools Appl. 60 (2012) 31–46, http://dx.doi.org/10.1007/s11042-011-0794-1.
[18] S. Arpa, S. Süsstrunk, R.D. Hersch, Revealing information by averaging, J. Opt. Soc. Amer. A 34 (2017) 743–751, http://dx.doi.org/10.1364/josaa.34.000743.
[19] N. Sahu, A. Sur, SIFT based video watermarking resistant to temporal scaling, J. Vis. Commun. Image Represent. 45 (2017) 77–86, http://dx.doi.org/10.1016/j.jvcir.2017.02.013.
[20] S.C. Chuang, C.H. Huang, J.L. Wu, Unseen visible watermarking, in: International Conference on Image Processing, ICIP, IEEE, 2007, pp. 261–264.
[21] T. Rabie, M. Baziyad, I. Kamel, Enhanced high capacity image steganography using discrete wavelet transform and the Laplacian pyramid, Multimed. Tools Appl. 77 (2018) 23673–23698, http://dx.doi.org/10.1007/s11042-018-5713-2.
[22] M.P.P. Kumar, B. Poornima, H.S. Nagendraswamy, C. Manjunath, A comprehensive survey on non-photorealistic rendering and benchmark developments for image abstraction and stylization, Iran J. Comput. Sci. 2 (2019) 131–165, http://dx.doi.org/10.1007/s42044-019-00034-1.
[23] R. Bala, G. Sharma, System optimization in digital color imaging, IEEE Signal Proc. Mag. 22 (2005) 55–63, http://dx.doi.org/10.1109/msp.2005.1407715.
[24] O. Juarez-Sandoval, E. Fragoso-Navarro, M. Cedillo-Hernandez, A. Cedillo-Hernandez, M. Nakano, H. Perez-Meana, Improved imperceptible visible watermarking algorithm for auxiliary informationdelivery, IET Biometrics 7 (2018) 305–313, http://dx.doi.org/10.1049/iet-bmt.2017.0145.
[25] A. Cedillo-Hernandez, M. Cedillo-Hernandez, F. Garcia-Ugalde, M. Nakano, H. Perez-Meana, A fast and effective method for static video summarization on a compressed domain, IEEE Lat. Am. Trans. 14 (2016) 4554–4559, http://dx.doi.org/10.1109/tla.2016.7795828.
[26] S. Rana, N. Sahu, A. Sur, Robust watermarking for resolution and quality scalable video sequence, Multimed. Tools Appl. 74 (2014) 7773–7802, http://dx.doi.org/10.1007/s11042-014-2023-1.
[27] A. Torralba, R. Fergus, W.T. Freeman, 80 Million tiny images: a large dataset for non-parametric object and scene recognition, IEEE Ttrans. Pattern Anal. Mach. Intell. 30 (2008) 1958–1970, http://dx.doi.org/10.1109/TPAMI.2008.128.
[28] X. Hou, J. Harel, C. Koch, Image signature: Highlighting sparse salient regions, IEEE Trans. Pattern Anal. Mach. Intell. 34 (2012) 194–201, http://dx.doi.org/10.1109/TPAMI.2011.146.
[29] A. Cedillo-Hernandez, M. Cedillo-Hernandez, M. Garcia-Vazquez, M. Nakano-Miyatake, H. Perez-Meana, A. Ramirez-Acosta, Transcoding resilient video watermarking scheme based on spatiotemporal HVS and DCT, Signal Process. 97 (2014) 40–54, http://dx.doi.org/10.1016/j.sigpro.2013.08.019.
[30] P. Yu, Y. Shang, C. Li, A new visible watermarking technique applied to CMOS image sensor, in: 8th International Symposium on Multispectral Image Processing and Pattern Recognition, SPIE, 2013, p. 8917, 2013.
[31] A. Phadikar, S.P. Maity, B. Verma, Region based QIM digital watermarking scheme for image database in DCT domain, Comput. Electr. Eng. 37 (2011) 339–355, http://dx.doi.org/10.1016/j.compeleceng.2011.02.002.
[32] A. Cedillo-Hernandez, M. Cedillo-Hernandez, M. Nakano-Miyatake, H. Perez-Meana, A spatiotemporal saliency-modulated JND profile applied to video watermarking, J. Vis. Commun. Image Represent. 52 (2018) 106–117, http://dx.doi.org/10.1016/j.jvcir.2018.02.007.
[33] Z. Wei, K.N. Ngan, Spatio-temporal just noticeable distortion profile for grey scale image/video in DCT domain, IEEE Trans. Circuits Syst. Video Technol. 19 (2009) 337–346, http://dx.doi.org/10.1109/tcsvt.2009.2013518.
[34] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error measurement to structural similarity, IEEE Trans. Image Process. 13 (2004) 600–612, http://dx.doi.org/10.1109/TIP.2003.819861.
[35] M.H. Pinson, S. Wolf, A new standardized method for objectively measuring video quality, IEEE Trans. Broadcast 50 (2004) 312–322, http://dx.doi.org/10.1109/TBC.2004.834028.
[36] A.K. Moorthy, A.C. Bovik, Efficient motion weighted spatiotemporal video SSIM index, in: Proceedings Human Vision and Electronic Imaging, IS & T/SPIE Electronic Imaging, SPIE, 2010, 75271I–75271I-9.
[37] Ffmpeg: A complete, cross-platform solution to record, convert and stream audio and video, 2021, https://www.ffmpeg.org/, (Accessed 08 April 2021).
[38] E.T. Lin, E.J. Delp, Temporal synchronization in video watermarking, IEEE Trans. Signal Process. 52 (2004) 3007–3022, http://dx.doi.org/10.1109/TSP.2004.833866.
[39] L. Agilandeeswari, K. Ganesan, A robust color video watermarking scheme based on hybrid embedding techniques, Multimed. Tools Appl. 75 (2016) 8745–8780, http://dx.doi.org/10.1007/s11042-015-2789-9.
[40] A. Bhardwaj, V.S. Verma, R.K. Jha, Robust video watermarking using significant frame selection based on coefficient difference of lifting wavelet transform, Multimed. Tools Appl. 77 (2017) 19659–19678, http://dx.doi.org/10.1007/s11042-017-5340-3.
[41] Q. Liu, S.Yang, J. Liu, P. Xiong, M. Zhou, A discrete wavelet transform and singular value decomposition-based digital video watermark method, Appl. Math. Model 85 (2020) 273–293, http://dx.doi.org/10.1016/j.apm.2020.04.015.
[42] I. Bayoudh, S.B. Jabra, E. Zagrouba, Online multi-sprites based video watermarking robust to collusion and transcoding attacks for emerging applications, Multimed. Tools Appl. 77 (2017) 14361–14379, http://dx.doi.org/10.1007/s11042-017-5033-y.
[43] M. El'Arbi, M. Koubaa, M. Charfeddine, C.B. Amar, A dynamic video watermarking algorithm in fast motion areas in the wavelet domain, Multimed. Tools Appl. 55 (2011) 579–600.
[44] T. Tabassum, S.M. Islam, A digital video watermarking technique based on identical frame extraction in 3-level DWT, in: 15th International Conferenceon Computer and Information Technology, ICCIT, IEEE, 2012, pp. 101–106.
[45] S. He, M. Wu, Collusion-resistant video fingerprinting for a large user group, IEEE Trans. Inf. Forensics Secur. 2 (2007) 697–709, http://dx.doi.org/10.1109/TIFS.2007.908179.
[46] K. Ait Sadi, A. Guessoum, A. Bouridane, F. Khelifi, Content fragile watermarking for H.264/AVC video authentication, Int. J. Electr. 104 (2016) 673–691, http://dx.doi.org/10.1080/00207217.2016.1242163.