Video Watermarking System for Copyright Protection based on Moving Parts and Silence Deletion

Shahad Almuzairai[1], Nisreen Innab[2]
Naif Arab University for Security Sciences, Department of Information Security
Ryadh, Saudi Arabia

***Abstract*—In recent years, video watermarking has emerged as a powerful technique for ensuring copyright protection. However, ensuring the lowest level of distortion, high transparency and transparency control, integrity of the watermarked video, and robustness against attacks that can be applied to destroy the embedded watermark are important properties that should be satisfied in a watermarking system. In this paper, we propose a video watermarking system that hides a watermark in both the visual and audio streams to ensure the integrity of the watermarked video. Specifically, we propose the moving block detection (MBD) algorithm for hiding the watermark in the moving parts of the original visual stream of the video. The MDB algorithm ensures that a minimal amount of distortion is caused by embedding the watermark. The MBD uses entropy to find the moving parts of the visual stream to hide the watermark. The process of hiding in the visual stream is performed using DWT to ensure both transparency and resistance against attacks. We employ the power factors of DWT to control the level of transparency. In addition, we propose the silence deletion algorithm (SDA), which generates a pure original audio stream by removing the noise from the original audio stream to form the hiding place of the watermark within the audio stream. DCT is employed to hide the watermark within the pure original audio stream to ensure resistance against attacks. Under a threat model, which includes bilinear, curved, and LPF geometric attacks and compression and Gaussian noise non-geometric attacks, the experimental results demonstrated that the proposed system outperformed four similar systems: key-frame-, I-frame-, spread-spectrum-, and LBS-based systems.**

Keywords*—Watermark; audio stream; visual stream; moving block, silence deletion; DWT; DCT; attacks

I. INTRODUCTION

Facing the ever-growing quantity of digital videos that are transmitted, shared and exchanged over the Internet, illegal copying and unreliable distribution of digital content have become serious, alarming problems.

Importance of video watermarking video watermarking can be defined as the process of hiding a watermark in a video [1, 2]. This watermark can be an image, audio, or text file. The importance of video watermarking is due to its valuable applications, such as authentication, tamper detection, and fingerprinting [3, 4, 5]. One of the most important applications of video watermarking is copyright protection [6, 7]. To demonstrate this feature, suppose that a company developed a special tool that contributes to resolving a critical issue. The solution is recorded by a video and transmitted via the Internet. To ensure the product ownership, the logo of the company is hidden within the video so that if an attacker tries to steal this product, the company can prove that this product is related to its own inventories by extracting the hidden logo.

Despite the benefits that are provided by video watermarking, it is not without problems. To define these problems, we must examine the general scenario of a video watermarking system, which is illustrated in Fig. 1.

Fig. 1 shows that the original video is manipulated to hide the original watermark. This process is called the embedding stage, which is performed at the sender side. At the receiver side, the contract process, which is called the extraction process, is executed; this yields the original video and the extracted watermark. Finally, the original watermark and the extracted watermark are matched to ensure the similarity.

Statement of the problem and the corresponding research questions. According to the previous Figure, embedding a digital watermark within a video ensures the copyright protection. However, the embedding process causes distortion of the original video. If this distortion is observed, the attacker can infer that this video is protected by a watermarking technique. Therefore, the original video (prior to the embedding process) must match the watermarked video (after the embedding process). Thus, the corresponding research question is as follows: How can the matching between the original video and the watermarked video be ensured? In addition, hiding a watermark within the video stream of the original video leads to an incomplete watermarking process because the video has another component (the audio stream) and the video file cannot be represented by only one part. This situation leads to the following research question: How can the accurate integration of the watermarked video be ensured? Moreover, the transparency of the embedded digital watermark, namely, the invisibility of the digital watermark to the naked human eye, is a critical issue and leads to the following research question: How can the transparency of the embedded digital watermark be ensured [8, 9]? Regarding ensuring transparency, another issue arises, which is related to controlling the level of the transparency that is realized after the video watermarking process. The corresponding research question is as follows: How can the transparency level be controlled to render the digital watermark invisible, semi-visible, or fully visible in the watermarked video [10, 11]? In addition, the attacker can manipulate the watermarked video by applying geometric or non-geometric attacks, such as a low-pass filter (LPF), rotation, compression, or noise addition [12, 13], which results in the destruction of the extracted digital watermark. The corresponding research question is as

follows: How can robustness against these types of attacks be ensured?

Motivated by the five research questions that are posed above, the construction of a robust video watermarking system that ensures copyright protection is essential.

By selecting a suitable location for the watermark to be hidden, we can ensure the matching between the original video and the watermarked video. In addition, employing frequency-based techniques, rather than spatial-based techniques such as least significant bit (LSB), endows the process of hiding with higher resistance against potential attacks.

The main contributions of this work are as follows:

- In response to the first three research questions, we propose a novel watermarking approach that ensures copyright protection while satisfying the requirements of video watermarking (no distortion and transparency). The process of hiding is performed in both the audio and visual streams. The no-distortion and transparency requirements are satisfied by hiding the watermark within the moving parts of the original video file with the help of the discrete wavelet transform (DWT). In the audio stream, the hiding process is performed using the discrete cosine transform (DCT).
- In response to the fourth research question, the transparency can be controlled (to high transparency or low transparency) in the proposed approach by adjusting the power factors of DWT.
- In response to the last research question, the proposed approach is resistant to various types of attacks, such as rotation, compression, LPF, salt and pepper, and Gaussian noise. The resistance is guaranteed in the video stream part by hiding the watermark within moving objects in the original video. Meanwhile, the resistance of the watermarked audio stream part is realized via a proactive silence-deletion-based step.

The remainder of the paper is organized as follows: Section II reviews the related works. Section III describes our proposed system, along with its components' roles, in detail. Security analysis is discussed in Section IV. Section V presents the metrics that were considered, followed by the experimental results and evaluations in Section VI. Finally, we present the conclusions of this work in Section VII.

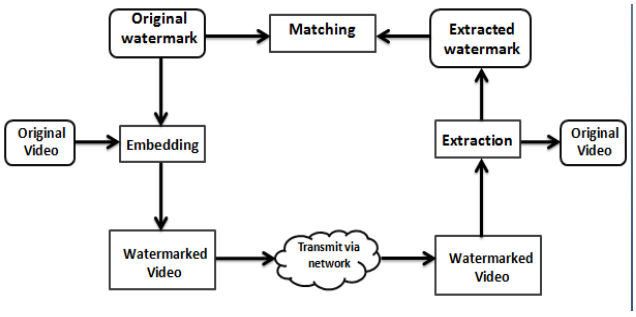


Fig. 1. General Scenario of a Video Watermarking System.

II. Related Work

Video watermarking approaches can be proposed under two main domains: the spatial domain and the frequency domain. Each domain has its own techniques, as illustrated in Fig. 2.

A. Spatial Domain

In this domain, a frame of the video (the image) is manipulated at the pixel level, where the color space is employed in the embedding process. The most common techniques that are used in this domain are reviewed below.

1) Additive watermarking technique: This technique focuses on the intensity of the pixels in the image, where the watermark will be hidden as a spread noise in terms of (-1, 0, +1) [14].

2) Least significant bit (LBS) technique: This technique is an old technique. Its key strategy is to hide the watermark within the least significant bit since it will produce the smallest distortion after hiding. Many enhancements over LSB can be applied, which involve encryption, randomization, or both. LSB can be used in both image and audio files [15].

3) Texture mapping coding technique: This technique is used only with noisy images. A noisy image is an image that contains many textured areas, which are the best places to hide the watermark [16].

4) SSM-modulation-based technique: This technique mainly utilizes spread-spectrum methods to modulate the color signal and embeds the watermark in the energy of the color wave [17].

The spatial-domain techniques are highly vulnerable to most attacks according to [18, 4]. Hence, the focus of research is moving toward the frequency domain.

B. Frequency Domain

In this domain, the color waves of the pixels are considered and the frame of the video is converted from the spatial domain to the frequency domain via mathematical transforms. The previous works can be classified into three main classes, as illustrated in Fig. 3.

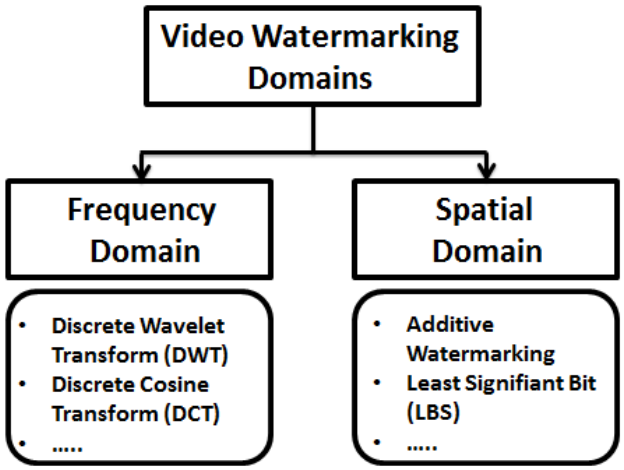


Fig. 2. Domains of Video Watermarking Approaches.

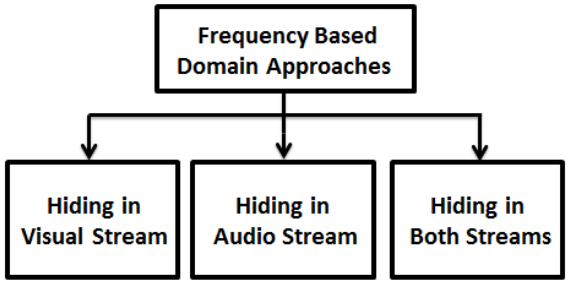


Fig. 3. Classification of Frequency-Based Domain Video Watermarking Approaches.

1) Hiding only in the visual stream of the video: In [19], the authors proposed a watermarking method in which the watermark is represented as a label and embedded in pixels of each frame via DCT. For this purpose, a search table of pixel patterns and their sign sequences of eight low DCT coefficients are exploited. The main advantage of this approach is that it is robust against changes in the group of pictures. Focusing on the transparency requirement, Ahmed at el. [20] proposed a blind video watermarking scheme. The watermark is embedded into preselected frames of the original video. Theses frames are selected based on a key value and are referred to as key frames. Then, the key frames are converted into the YUV color system and the watermark is hidden in the luminance layer (Y layer) using DWT to ensure transparency. To make the process blind, the watermarked video was manipulated without the original video, where the key frames are manipulated using the inverse DWT to extract the hidden watermark. This approach provides high invisibility of the watermark and requires less processing time compared to the previous approach since the hiding process is not applied on all frames of the original video. However, the process of selecting key frames may not be suitable for many video files formats.

Another watermarking method is presented in [21], which uses static 3D-DCT to hide a watermark in video. The key strategy is to identify a scene change in the video and convert the frames into the YUV color space to select the luminance layer (Y) for the hiding process. This model yields satisfactory results for videos that have low motion activity; in other cases, there is noticeable distortion. Similar to the previous work, the authors of [22], who developed the previous model, used dynamic 3-D DCT to realize the benefits of utilizing the frequency of the video sequences, which provides more robustness against attacks.

In [23], a copyright video protection approach is proposed. DWT is used in the hiding process, where it is implemented on both the watermark and the I-frames that represent the location for hiding. Instead of converting the I-frames from RGB into YUV, the authors use the YCbCr color space to realize the transparency objective. This work was subsequently enhanced by the same authors, who focused on capacity and security features [24]. The capacity feature is realized by manipulating the original video at the bit level, while the security feature is realized by encrypting the watermark prior to hiding it.

2) Hiding only in the audio stream of the video: Based on an audio stream compression method, Petrovic et al. proposed an audio stream watermarking approach [25]. They focused on minimizing the processing requirements at the embedding side while maintaining high perceptual quality. The key strategy is to employ advanced audio coding (AAC) technology. Two main steps are performed: (1) preprocessing and (2) marking. In the preprocessing step, a host signal is marked by one or more hiders. Each hider embeds a string of identical symbols. In the second step, two or more distinct copies of the host signal are retrieved from the memory to be input to a multiplexer (MUX) when the creation of a marked copy is requested. However, this approach has a substantial drawback: it is vulnerable to compression non-geometric attacks.

The authors of work [26] were motivated to deal with the audio stream because due to the narrow-bandwidth limitation, speech signals are seldom used, despite their popularity in communication applications, such as military, bank, phone and network security. Therefore, they proposed a spread-spectrum-based technique for hiding the watermark within the audio steam. The authors combine direct-sequence spread spectrum (DSSS) technology with a simple basic frequency mask to conduct the hiding process.

In [27], a three-step audio watermarking system is proposed. The first step is to use the standard LBS technique. The second step is to search for the level of audio that is closest to the level of the original audio after watermarking. The search process depends on the minimum error level. The main objective of the second step is to ensure transparency. The third step utilizes error diffusion to ensure the high capacity of the proposed system.

To realize high capacity when hiding data in the audio signal, the authors of [28] utilized the fast Fourier transform (FFT) spectrum. The key strategy is to divide the FFT spectrum into short frames and change the magnitudes of selected FFT samples using Fibonacci numbers. Using Fibonacci numbers, it is possible to change the frequency samples adaptively.

3) Hiding in both the visual and audio streams of the video: A self-adaptive approach is proposed in [29] for hiding a watermark within both the visual and audio streams. The authors relied on two main processing steps: The watermark is constructed from the audio stream of the video, where the features of the audio signal are extracted and used to generate the watermark. Then, the generated watermark is embedded within the visual stream via DCT.

Aiming at providing a solution with robust and fragile aspects to guarantee authentication and integrity, the authors of [30] proposed an approach that uses watermarks in combination with content information. The authors used the same strategy as in the previous work. The main difference is that they used a seed-based method in the hiding process.

III. Proposed System

In this section, we introduce our proposed video watermarking system, which satisfies the integrity, transparency, and robustness requirements. The section is organized as follows: a threat model is defined, followed by the corresponding architecture of the proposed video watermarking system. Then, the role of each component of the system architecture is described in detail.

A. Threat Model

In the context of defining the threat model, we define the attacker, his/her objective, the type of the attack, and the capabilities of the attacker that are used to achieve the objective.

For an original video (O_{video}) with both a visual stream (OV_{stream}) and an audio stream (OA_{stream}), (OV) is defined as:

$$O_{video} = OV_{stream} \cup OA_{stream} \quad (1)$$

After hiding the original watermark (OW) within both OV_{stream} and OA_{stream}, a watermarked video (W_{video}) is generated as:

$$W_{video} = WV_{stream} \cup WA_{stream} \quad (2)$$

where

$WV_{stream} = \bigcup_{OV_{stream}}^{OW}$ and $WA_{stream} = \bigcup_{OA_{stream}}^{OW}$

The type of the attack is active. Therefore, the objective of the attacker (man in the middle) is to destroy the embedded watermark, as illustrated in Fig. 4.

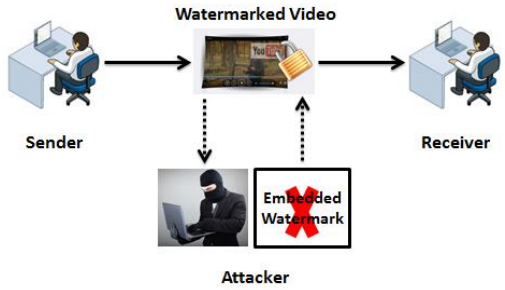


Fig. 4. Objective of the Attacker.

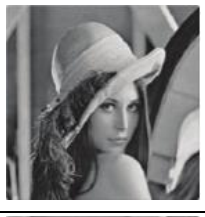
To accomplish his/her objective, the attacker uses geometric or non-geometric attacks. Table 1 lists the capabilities of the attacker.

Table 2 shows the effects of the previously described attacks on an image (or video frame).

TABLE I. Capabilities of the Attacker

Cap NO	Attack Type	Original Video Streams: Visual Stream	Original Video Streams: Audio Stream
1	Geometric Attacks	Bilinear	×
2		Curved	×
3		LPF	×
4	Non-geometric attacks	Compression	Compression
5		Gaussian Noise	Gaussian Noise

TABLE II. Effects of Attacks

Attack Name	Original Image	Effect
Bilinear		
Curved		
LPF		
Gaussian Noise		
Compression		

B. Our Proposed System Architecture

The framework of the proposed system consists of the sender and the receiver of the watermarked video and the attacker. All three are connected via a network. The system is managed by eight components ($Recorder_{OV}$, $Splitter$, $Finder_{HP}$, $Hider_{DWT}$, $Remover_S$, $Hider_{DCT}$, $Adder_S$ and $Mixer$), as shown in Fig. 5.

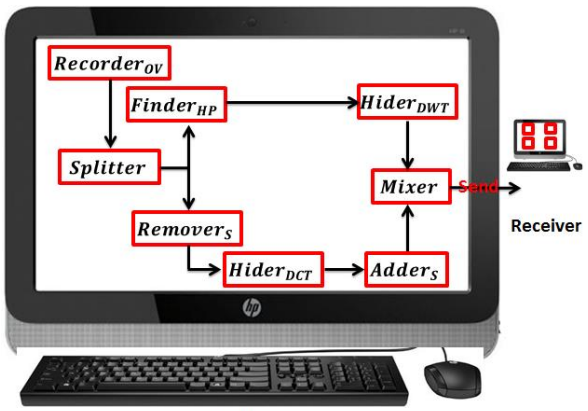


Fig. 5. Our Proposed System Architecture.

TABLE III. COMPONENTS

Name	Main mission	Location
$\text{Recorder}_{\text{OV}}$	Recording the original video.	Sender side.
Splitter	Extracting original visual and audio streams.	Sender & receiver sides.
$\text{Finder}_{\text{HP}}$	Finding the place of hiding within the visual stream.	Sender & receiver sides.
$\text{Hider}_{\text{DWT}}$	Hiding process within the visual stream.	Sender & receiver sides.
$\text{Remover}_{\text{S}}$	Deleting silence.	Sender & receiver sides.
$\text{Hider}_{\text{DCT}}$	Hiding process within the audio stream.	Sender & receiver sides.
Adder_{S}	Adding silence.	Sender & receiver sides.
Mixer	Merging the watermarked visual and audio streams.	Sender side.

Table 3 lists the components and identifies the main mission of each component and where it is installed.

The mission of each component is integrated with the missions of the others. The following explains the roles of the components.

C. Roles of the Components

1) Role of the $Recorder_{OV}$ component: This component is responsible for creating the original video (both the visual and audio streams). Any multimedia recorder can be used here; the generated video file can be converted later into other formats. We used the Zoom program for this purpose [32].

2) Role of the Splitter component: This component is responsible for obtaining the visual and audio streams of the recorded original video separately. At the end, the two streams are ready for the hiding process. We use the Wondershare Filmora multimedia tool for this purpose [33].

3) Role of the $Finder_{HP}$ component: This component is responsible for identifying a suitable place for the original watermark to be embedded. Selecting the suitable place to hide the original watermark mainly contributes to ensuring matching between the original video and the watermarked one. The $Finder_{HP}$ component executes the moving part detection approach (MPDA), as described below.

D. Moving Part Detection Approach (MPDA)

Randomly selecting a part of a frame for hiding is a poor solution because, depending on the static parts of a frame, for example, leads to highlighting of the distortion after the watermark has been hidden. By contrast, depending on moving parts of the frame is an effective strategy for hiding because the moving parts of the frame can be viewed as a type of noise, which is referred to as the dirty window effect [31], which is demonstrated in Fig. 6.

In Fig. 6, two frames of a Miss America contestant are shown, in which the woman is speaking. In the frames within a video, the moving part (i.e., her mouth) appears as a noise. Inserting a watermark leads to distortion, where foreign information is added to the pure visual stream of the original video. However, inserting a watermark within such a moving part will not lead to a noticeable change. The reason behind this is that the result of the insertion process can be viewed as a noise over a noise. This, in turn, leads to unnoticeable distortion, which contributes to the matching of the original visual stream with the watermarked one.

The $Finder_{HP}$ component separates the original frame into moving and non-moving parts, as illustrated in Fig. 7.

To identify the blocks of the moving parts from the original visual stream (rather than the non-moving parts), we utilize the entropy metric. In image processing, entropy is used to classify textures: a texture might correspond to a known entropy value if patterns repeat themselves in approximately regular ways, which is true in videos in which the frames are periodically repeated to create the motion.

Specifically, the watermark is embedded in the moving part of each color frame in all three RGB channels. Several beginning frames of original visual stream are selected as references. Then, the state of each block that is involved in the current frame is determined (moving or non-moving), which is accomplished by comparing the entropy value of each block (in the current frame) with the corresponding entropy values of the references blocks. If the difference between the entropy values is high, a high disorder or high variance is detected. Thus, the current block is moving; otherwise, it is non-moving. Entropy has already been implemented as a function in Matlab. Fig. 8 illustrates this strategy.

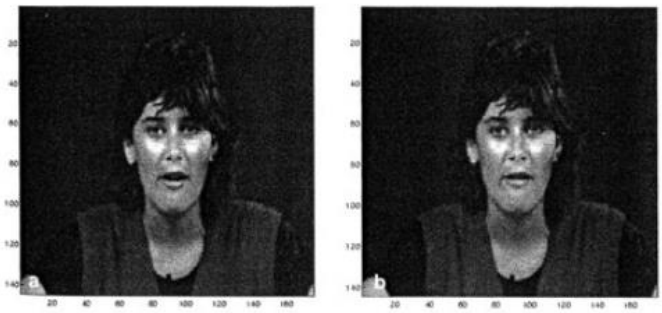


Fig. 6. Dirty Window Effect.

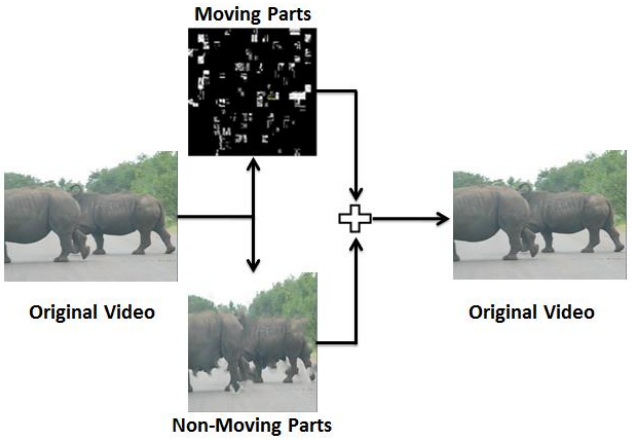


Fig. 7. Moving and Non-Moving Parts of an Original Video.

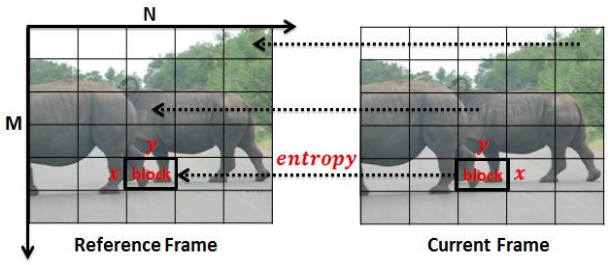


Fig. 8. Moving and Non-Moving Block Detection.

Formally, each $m \times n$ color channel is divided into blocks of size $x \times y$. Let $M = \frac{m}{x}$ and $= \frac{n}{y}$. Then, each block can be represented as:

$$Block_{ij}^{c} \quad i \in \{1,2,\dots M\}, \; j \in \{1,2,\dots N\} \tag{3}$$

where $c = \{R, G, B\}$

To accurately determine the entropy value, which will be used to decide whether a block is moving or non-moving, we use a normalization process. The average of all entropy values from all blocks is calculated as:

$$AVE_c = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} E\,(Block_{ij}^{c}) \tag{4}$$

where E denotes the entropy function.

Any block can be evaluated as moving or non-moving as follows:

$$(Block_{ij}^{c}) \; is \; \begin{cases} moving \; if \; (e \geq AVE_c) \\ non-moving \; if \; (e < AVE_c) \end{cases} \tag{5}$$

where (e) denotes the entropy value of the specified block.

Algorithm 1 presents the pseudo code of the mission of the $Finder_{HP}$ component.

Algorithm 1: Moving Block Detection (MBD)

Input: Frames of the original visual stream, x, y, ref-frames.
Output: Moving-blocks-array[]. Moving blocks of each frame.

```
Ref-F-Array [] = ∅;
Moving-blocks-array[] = ∅;
for (ref=1; ref <= ref-frames; ref ++)
    add frame to Ref-F-Array [];
end for
for (d=1; d <= ref-frames; d ++)
    for (i=1; i <= M; i ++)
        for (j=1; j <= N; j ++)
            cut block of size (M/x, N/y);
            calculate the entropy of the block
        end for
    end for
end for
calculate AVE_c;
if block entropy > AVE_c then
add block to Moving-blocks-array[];
return Moving-blocks-array[];
```

1) Role of the $Hider_{DWT}$ component: This component is responsible for hiding the original watermark within the moving blocks that are obtained from the executed mission of the previous component. The mission of the $Hider_{DWT}$ component is performed using DWT. In addition, it makes it possible to control the transparency of the embedded watermark.

E. DWT-Based Hiding Approach (DWTHA)

By definition, DWT generates a sparse time–frequency representation of an input signal. The output of DWT is four subbands of data: a low/low-frequency band (LL), a low/high frequency band (LH), a high/low frequency band (HL), and a high/high frequency band (HH) [34]. Most of the information of the input signal is included in LL subband and the other subbands are viewed as shadows of the input signal that have decreased appearance quality, which gives DWT an advantage: multi-resolution. The key power of the multi-resolution feature is that the localization characteristics match the theoretical models of the human visual system (HVS). Depending on the localization characteristics of the multi-resolution feature, a watermark can be embedded within any of the four generated subbands. However, embedding a watermark within the HH subband results is a high transparency requirement guarantee, but leads to low resistance against attacks. Meanwhile, embedding a watermark within the LL subband results in high resistance against attacks but leads to noticeable distortion (thereby decreasing the quality of the watermarked signal) [35].

To solve this problem, the moving blocks are converted from the RGB color system into the YUV color system. Then, the Y layer, which refers the luminance layer, is extracted. Finally, DWT is applied on the Y layer and the watermark is embedded within the LL subband, as illustrated in Fig. 9.

As illustrated in Fig. 9, the hiding process is performed within the Y layer of the detected moving block. Both the transparency of the embedded watermark and the resistance against attacks are ensured by hiding each resultant subband in the corresponding subband.

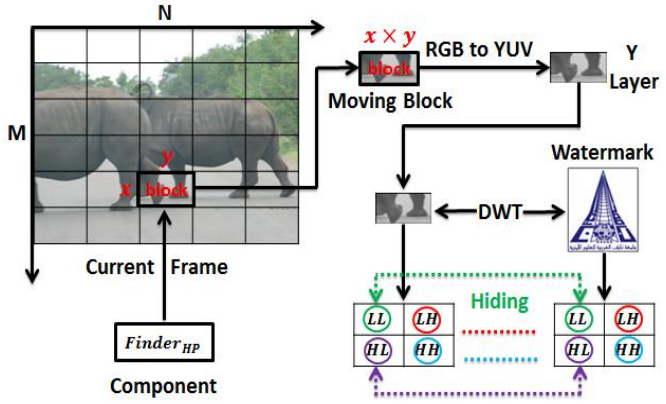


Fig. 9. Hiding Process.

Formally, a one-dimensional DWT is expressed as:

$$W(k,j) = \frac{1}{\sqrt{M}} \sum_x f(x) 2^{\frac{j}{2}} \sigma\,(2^j x - k) \quad (6)$$

$$\psi = \begin{cases} 1, & 0 \le x \le 0.5 \\ -1, & 0.5 \le x \le 1 \\ 0, & otherwise \end{cases} \quad (7)$$

where W represents the wavelet coefficient function; j and k denote the dilation and translation parameters, respectively; and M is the length of the signal f.

For images (i.e., frames), two-dimensional DWT is used. Two-dimensional DWT is derived from one-dimensional DWT. A two-dimensional scaling function and three-dimensional wavelets are required, as follows:

$$\rho(x,y) = \rho\,(x) \times \rho(y) \quad (8)$$

$$\sigma^X(x.y) = \sigma\,(x) \times \rho(y) \quad (9)$$

$$\sigma^Y(x.y) = \rho(x) \times \sigma\,(y) \quad (10)$$

$$\sigma^Z(x.y) = \sigma\,(x) \times \sigma\,(y) \quad (11)$$

The expanded and translated basis functions are:

$$\rho_{j,m,n}\,(x,y) = 2^{\frac{j}{2}}\,\rho\,(2^j x - m, 2^j x - n) \quad (12)$$

$$\sigma_{j,m,n}\,(x,y) = 2^{\frac{j}{2}}\,\rho\,(2^j x - m, 2^j x - n) \quad (13)$$

where $i = \{X, Y, Z\}$

Then, the discrete wavelet transform function $f\,(x,y)$ of size $M \times N$ is:

$$W_\rho(m.n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \rho_{m,n}(x_0) \quad (14)$$

$$W_\sigma(m.n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \sigma_{m,n}(x_0) \quad (15)$$

The two previous formulas are applied in the luminance layer (Y) of the moving blocks ($Block_{ij}^c$), where (1) each block is of size $(x \times y)$ and (2) $R = X, G = Y,$ and $B = Z$. Thus, DWT decomposes the two-dimensional moving block into wavelet-like matrices (i.e., the four subbands that are illustrated in Fig. 9). In addition, DWT decomposes the original watermark into the four corresponding subbands.

Let ll_w, hl_w, lh_w, and hh_w denote the four subbands that represent the output of DWT on the original watermark, which is denoted as W_O. Let ll_B^O, hl_B^O, lh_B^O, and hh_B^O denote the corresponding subbands of a moving block that was extracted from OV_{stream}. The hiding process is performed according the following formulas:

$$\begin{cases} ll_B^W = (1 + \mu_1 \times \mathrm{ll_w}) \times ll_B^O \\ hl_B^W = (1 + \mu_2 \times \mathrm{hl_w}) \times hl_B^O \\ lh_B^W = (1 + \mu_3 \times \mathrm{lh_w}) \times lh_B^O \\ hh_B^W = (1 + \mu_4 \times \mathrm{hh_w}) \times hh_B^O \end{cases} \quad (16)$$

where ll_B^W, hl_B^W, lh_B^W, and hh_B^W denote the watermarked subbands of the moving block. The coefficient vector μ_{TC} $(TC = 1, 2, 3, 4)$ contains the power factors that are related to the transparency. This vector is used to control the transparency value of the embedded watermark, where $\mu_{TC} \in]0, 1[$. If μ_{TC} has high values, then the embedded watermark is visible in the watermarked video (i.e., poor transparency). If μ_{TC} has low values, then the embedded watermark is invisible in the watermarked video (i.e., satisfactory transparency). Thus, by adjusting the values of the power factors, full control of the embedded watermark can be realized (visible, invisible, and semi-visible).

Algorithm 2 presents the pseudocode of the mission of the $Hider_{DWT}$ component.

Algorithm 2: DWT-based Hiding Process.

Input: Moving-blocks-array[],$W_O$original watermark, μ_{TC} values.
Output: Watermarked Moving-blocks.
1: read W_O;
2: $rgb2gray(W_O)$;
3: DWT (W_O);
4: **call** MBD function (OV_{stream});
5: **while** size (Moving-blocks-array[] $\neq \emptyset$) do
6: $rgb2gray(\mathrm{Moving-blocks-array[]})$;
7: extract Y (luminance) layer;
8: DWT (Y layer of blocks);
9: $ll_B^W = (1 + \mu_1 \times ll_w) \times ll_B^O 7$;
10: $hl_B^W = (1 + \mu_2 \times hl_w) \times hl_B^O$;
11: $lh_B^W = (1 + \mu_3 \times lh_w) \times lh_B^O$;
12 $hh_B^W = (1 + \mu_4 \times hh_w) \times hh_B^O$;
13: **end while**
14: return Watermarked-blocks-array[];

1) Role of the Remover$_S$ component: This component is responsible for manipulating the original audio stream to prepare it for the hiding process. This manipulation is performed in a pre-processing stage via the silence deletion approach, as described below.

F. Silence Deletion Approach (SDA)

Typically, speech signals vary slowly over time. Therefore, if a speech signal is detected over a short time window, it reflects stationary characteristics (i.e., silence parts). Meanwhile, if it is detected over a long time window, it reflects changing characteristics, which lead to various speech sounds. Typically, the first 200 msec of a speech signal (approximately 1600 samples) correspond to the silence parts. In addition, the silence parts can spread over a speech signal [36], as shown in Fig. 10.

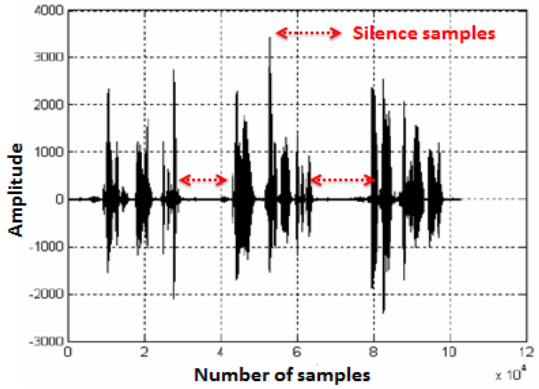


Fig. 10. Silence Samples of a Speech Signal.

The key strategy for preparing an audio stream for the hiding process is to detect and delete the silence samples so that the watermark is embedded within the pure original audio stream. This strategy can provide high resistance against compression attacks because the compression attacks delete the silence samples to decrease the size of an audio file. Thus, if a compressions attack is applied on a watermarked audio stream, the embedded watermark will not be affected.

Formally, let ζ and λ denote the mean and standard deviation, respectively, of the first 1600 samples (ϑ) of an original audio stream. Then, the noise that is distributed over the audio signal is expressed as:

$$\zeta = \frac{1}{600} \times \sum_{s=1}^{1600} \vartheta(s) \quad (17)$$

$$\lambda = \sqrt{\frac{1}{600} \times \sum_{s=1}^{1600} (\vartheta(s) - \lambda)^2} \quad (18)$$

A sample ϑ is categorized as silence or voiced via the following formula:

$$\vartheta \begin{cases} voiced, \ if \ (\frac{|\vartheta - \zeta|}{\lambda} < 3) \\ silence, \ othersise \end{cases} \quad (19)$$

To represent the original audio signal as a series of zeros and ones, we label the voiced samples as ones and the silence samples as zeros. Thus, the audio signal is decomposed into two non-overlapping windows of voiced and silence samples. The process of marking the silence samples consists of two steps: (1) labeling the silence samples and (2) associating the label with the location of the silence sample. Via these two steps, the silence part is obtained, saved and, finally, deleted from the original audio stream. Later, we reincorporate the silence part after watermarking the original pure audio stream.

Algorithm 3 presents the pseudocode of the mission of the $Remover_S$ component.

Algorithm 3: Silence Deletion Approach

Input: Original audio stream (OA_{stream})
Output: Pure-audio-stream [], hash of silence samples ($hash\ [key = position, value = duration]$)

1: FS = read ($OA_{stream}[1:1600]$);
2: $\zeta = average\ (FS)$;
3: $\lambda = deviation\ (FS)$;
4: pure-c=0;
5: **for** (s=1; ds<= $length\ (OA_{stream})$; s ++)
6: **if** ($\frac{|OA_{stream}(s) - \zeta|}{\lambda} < 3$) then
7: duration =0;
8: **while** ($\frac{|OA_{stream}(s) - \zeta|}{\lambda} < 3$) do
9: duration = duration +1;
10: s=s+1;
11: **end while**
12: $hash\ [s] = duration$;
13: **end if**
14: **else**
15: pure-c= pure-c+1;
16: Pure-audio-stream [pure-c] = $OA_{stream}(s)$;
17: **end else**
18: **end for**
19: return Pure-audio-stream, $hash$;

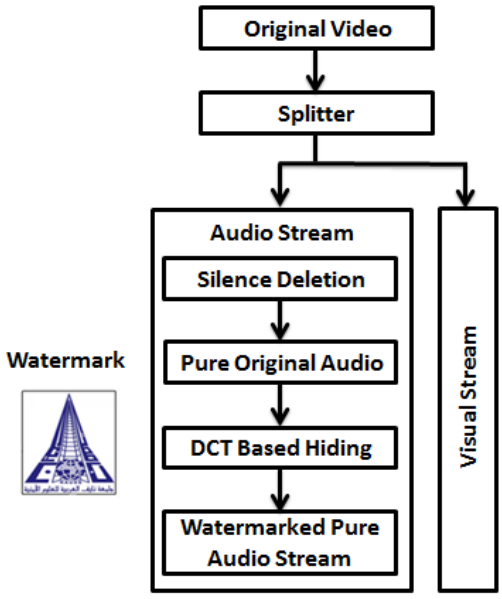


Fig. 11. Hiding within an Audio Stream.

1) Role of the $Hider_{DCT}$ component: This component is responsible for hiding the original watermark within the pure original audio stream. Here, the process of hiding mainly depends on DCT. The hiding process is performed on the pure original audio stream after silence samples have been deleted, as illustrated in Fig. 11.

Formally, let POA_{stream} and $S_{deletion}$ denote the pure original stream and silence samples, respectively. Then,

$$POA_{stream} = OA_{stream} - S_{deletion} \quad (20)$$

The watermark is embedded within the POA_{stream} by modifying the DCT coefficients. DCT is formulated as:

$$F(k) = coff(k) \times \sum_{P=0}^{P} f(P) \times cos[\frac{(2 \times P+1) \times k \times \pi}{2 \times N}] \quad (21)$$

$$(P \geq 0 \ and \ k < N)$$

where f(P) is the time pure original audio stream series, F(k) are the DCT coefficient series, and N is the number of samples on which DCT is performed.

Inverse DCT (IDCT) is expressed as:

$$f(P) = \sum_{k=0}^{N-1} coff(k) \times F(k) \times cos[\frac{(2 \times P+1) \times k \times \pi}{2 \times N}] \quad (22)$$

where $coff(k)$ is a coefficient that is defined as follows:

$$coff(k) = \begin{cases} \frac{1}{\sqrt{N}}, k = 0 \\ \sqrt{\frac{2}{N}}, k \neq 0 \end{cases} \quad (23)$$

When a watermark $W(i)$ is embedded within the i^{th} DCT coefficient, the i^{th} coefficient is modified:

$$\widetilde{F(i)} = F(i) + W(i) \quad (24)$$

Then, the corresponding time series are obtained via the IDCT as follows:

$$\widetilde{f(P)} = \sum_{k=0}^{N-1} coff(k) \times \widetilde{F(k)} \times \cos\left[\frac{(2 \times P + 1) \times k \times \pi}{2 \times N}\right]$$

$$= \sum_{k=0}^{N-1} coff(k) \times F(k) \times \cos\left[\frac{(2 \times P + 1) \times k \times \pi}{2 \times N}\right] +$$

$$coff(i) \times W(i) \times \cos\left[\frac{(2 \times P + 1) \times i \times \pi}{2 \times N}\right]$$

$$= f(P) + Noise\ (i, P) \quad (25)$$

where

$$Noise\ (i, P) = coff(i) \times W(i) \times \cos\left[\frac{(2 \times P + 1) \times i \times \pi}{2 \times N}\right] \quad (26)$$

Noise (i, P) represents the noise that is caused by the modification of the i[th] DCT coefficient on the P[th] sample in the time domain.

2) Role of the $Adder_S$ component.: This component is responsible for adding back the silence samples that are saved in the hash that was used in the silence deletion approach. Therefore, the input of this component is the watermarked pure audio stream and the output is the watermarked audio stream (WA_{stream}), as illustrated in Fig. 12.

3) Role of the Mixer component: This component is responsible for combining the watermarked visual stream (WV_{stream})and the watermarked audio stream (WA_{stream}), as inputs, to produce the watermarked video (W_{video}) as output, as illustrated in Fig. 13.

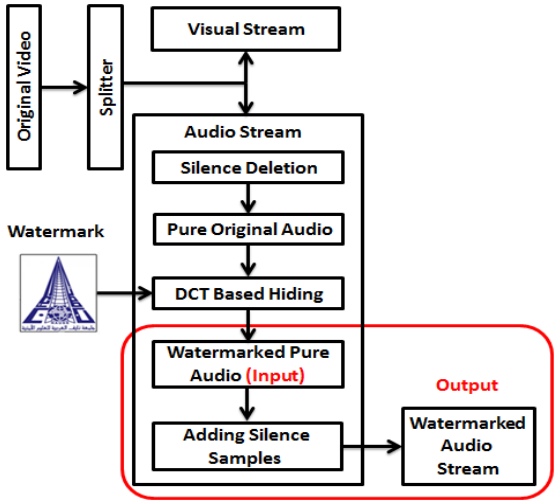


Fig. 12. Generating a Watermarked Audio Stream.

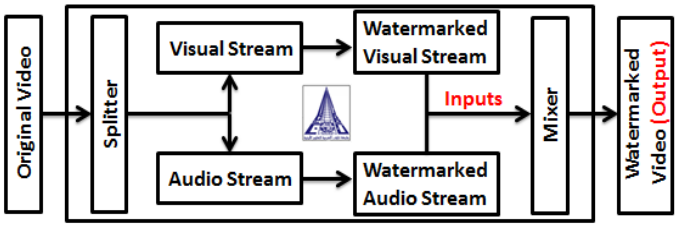


Fig. 13. Generating a Watermarked Video.

IV. Security Analysis

In this section, we prove that the attacks considered in the threat model fail to destroy the embedded watermark. We follow the definition-theorem-proof style in discussing the resistance against both geometric and non-geometric attacks.

A. Security Analysis of Geometric Attacks

Definition 1. A video watermarking system is bilinear attack resistant if the boundaries of the host frame (or image) do not change differently (in length or direction) such that the embedded watermark can be distinguished.

Theorem 1. The proposed video watermarking system is bilinear attack resistant.

Proof 1. Let $O_{im}(h, w, \theta_1, \theta_2, \theta_3, \theta_4)$ denote the original image (or frame) where the watermark is hidden, where $h, w, \theta_1, \theta_2, \theta_3,$ and θ_4 represent the height, width, and four boundary angles (i.e., properties), respectively. After the bilinear attack has been applied, the resultant (distorted) image will be $D_{im}^{..}(\ddot{h}, \ddot{w}, \ddot{\theta_1}, \ddot{\theta_2}, \ddot{\theta_3}, \ddot{\theta_4})$. Due to the motion, the moving parts of O_{im} be distorted (i.e., updating the properties). This distortion can be represented as $D_{im}^{.}(\dot{h}, \dot{w}, \dot{\theta_1}, \dot{\theta_2}, \dot{\theta_3}, \dot{\theta_4})$. Since the watermark is embedded within the moving parts of O_{im}, the distortion that is caused by the hiding process is:

$$\begin{Bmatrix} \ddot{h} \\ \ddot{w} \\ \ddot{\theta_1} \\ \ddot{\theta_2} \\ \ddot{\theta_3} \\ \ddot{\theta_4} \end{Bmatrix} = \begin{Bmatrix} h - \dot{h} \\ w - \dot{w} \\ \theta_1 - \dot{\theta_1} \\ \theta_1 - \dot{\theta_1} \\ \theta_1 - \dot{\theta_1} \\ \theta_1 - \dot{\theta_1} \end{Bmatrix} \quad (27)$$

The distortion is sufficiently small to preserve the features of the embedded watermark. Hence, the bilinear attack fails.

Definition 2. A video watermarking system is curved attack resistant if the boundaries of the host frame do not change equally (in an arc manner) such that the watermark can be distinguished.

Theorem 2. The proposed video watermarking system is curved attack resistant.

Proof 2. The same justification as was provided for the bilinear attack can be provided here, while taking into consideration the effect of the curved attack. That is because the effect of the curved attack is similar to that of the bilinear attack, with different property values of the resultant frame. Therefore, hiding within moving parts of the video contributes to the failure of the curved attack.

Definition 3. A video watermarking system is LBF attack resistant if the smoothness of the host frame does not change substantially such that the watermark can be distinguished.

Theorem 3. The proposed video watermarking system is LPF attack resistant.

Proof 3. Let $O_{im}(h, w, \theta_1, \theta_2, \theta_3, \theta_4, Smooth_{NL})$ denote the original image (or frame) where the watermark is hidden, where $h, w, \theta_1, \theta_2, \theta_3,$ and θ_4 represent the height, width, and

four boundary angles (i.e., properties), respectively, and suppose the smoothness is at a natural level. After applying the LPF attack, the resultant (distorted) image is denoted as $D_{im}(h, w, \theta_1, \theta_2, \theta_3, \theta_4, Smooth_{CL})$, where $Smooth_{CL}$ denotes the changed smoothness level. The smoothness level of the moving parts of the host frame was originally natural due to the motion ($Smooth_{NCL}$). Consequently, $Smooth_{NCL}$ is considered a part of $Smooth_{CL}$ that is caused by the LPF attack. Therefore, the watermark is embedded within the frame that has $Smooth_{NCL}$, which, in turn, mitigates the effect of the LPF attack since it can be viewed as a distortion over a distortion. In other words, a part of the effect of the LPF attack ($Smooth_{CL}$) is absorbed by $Smooth_{NCL}$. Hence, this feature of the host frame is preserved and the embedded watermark is not altered. As a result, the LPF attack fails.

B. Security Analysis of Non-Geometric Attacks

Definition 4. A video watermarking system is Gaussian noise attack resistant if the resolution of the pixels in the host frame does not decrease substantially such that the watermark can be distinguished.

Theorem 4. The proposed video watermarking system is Gaussian noise attack resistant.

Proof 4. Let $O_{im}(h, w, \theta_1, \theta_2, \theta_3, \theta_4, Res_{px})$ denote the original image (or frame) where the watermark is hidden, where $h, w, \theta_1, \theta_2, \theta_3$, and $\theta_4$represent the height, width, and boundary angles (i.e., properties), respectively. The first six properties are not affected by the Gaussian noise attack and we examine the change in the resolution due to the added noise. After applying the Gaussian noise attack, the resultant (distorted) image is denoted as $D_{im}(h, w, \theta_1, \theta_2, \theta_3, \theta_4, \widetilde{Res_{px}})$, where $\widetilde{Res_{px}}$ denotes the new resolution. When adding the Gaussian noise to the moving parts of the host frame, it is viewed as a noise over a noise since the motion itself can be viewed as a type of noise, which changes the resolution of the frame when it is viewed by human eyes. Therefore, the Gaussian noise is also absorbed by the noise of the motion. In other words, the embedded watermark is inserted within the noisy part of the host frame, which, in turn, prevents the Gaussian noise attack from destroying the watermark. In the audio stream, the Gaussian noise attack also fails because the watermark is embedded within the pure audio stream and is not substantially affected by this attack; the silence that is deleted is considered to be the place where the noise of the Gaussian attack is added.

Definition 5. A video watermarking system is compression attack resistant if both the resolution and contrast of the pixels in the host frame do not increase such that the watermark can be distinguished.

Theorem 5. The proposed video watermarking system is compression attack resistant.

Proof 5. Let $O_{im}(h, w, \theta_1, \theta_2, \theta_3, \theta_4, Res_{px}, Cont_{px})$ denote the original image (or frame) where the watermark is hidden, where $h, w, \theta_1, \theta_2, \theta_3$, and $\theta_4$denote the height, width, and four boundary angles (i.e., properties), respectively, and Res_{px} and $Cont_{px}$denote the resolution and the contrast. After applying the compression attack, the resultant (distorted) image will be $D_{im}(h, w, \theta_1, \theta_2, \theta_3, \theta_4, \overline{Res_{px}}, \overline{Cont_{px}})$, where $\overline{Res_{px}}$ refers to the new resolution and $\overline{Cont_{px}}$ refers to the new contrast. Since most of the representation of the watermark is embedded within the LL subband of the Y layer of the moving parts, the new resolution does not affect the embedded watermark. Moreover, because the shadows of the watermark are embedded within the corresponding shadows of the moving parts of the Y layer, the new contrast does not affect the embedded watermark. Therefore, the strategic employment of the multi-resolution feature of DWT contributes to the failure of the compression attack. Regarding hiding in the audio stream, the effect of the compression attack will be limited within the space of silence that was originally deleted before the hiding process. Therefore, the space of hiding (i.e., the pure audio) is not affected and the hidden watermark is kept safe.

V. METRICS

To evaluate the proposed video watermarking system, several metrics are used to measure the quality of video (QoV) after watermarking and the similarity between the original watermark and the extracted one.

A. QoV Metrics

To evaluate the QoV, the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) metrics are used. Calculating the PSNR value requires two inputs: a frame from the original video and a frame from the watermarked video. Let ($\mathrm{F_o}$) and ($\mathrm{F_w}$) refer the original frame and the corresponding watermarked frame, respectively, both of which are of size (M × N). Then, the PSNR is represented by:

$$\mathrm{PSNR} = 10 \times \log_{10} \frac{255^2}{\mathrm{MSE}(\mathrm{F_o}, \mathrm{F_w})} \tag{28}$$

where the mean squared error (MSE) is given by:

$$\mathrm{MSE} = \frac{1}{\mathrm{M} \times \mathrm{N}} \sum_{\mathrm{m}=1}^{\mathrm{M}} \sum_{\mathrm{n}=1}^{\mathrm{N}} [\mathrm{F_o} - \mathrm{F_w}]^2 \tag{29}$$

A higher PSNER value corresponds to a satisfactory QoV. A lower MSE value also corresponds to a satisfactory QoV, where the optimal QoV is obtained when the MSE value is close to zero.

The SSIM metric is used to quantify image quality degradation and to accurately measure the variation of structural information between the original frame (F_o) and the watermarked Frame (F_w). SSIM is defined in the context of three components: the luminance, contrast, and structural components. Formally, it is defined as:

$$\mathrm{SSIM}(W_o, W_{ext}) = [\mathrm{lum}\ (F_o, F_w)]^{\psi} \times [\mathrm{con}\ (F_o, F_w)]^{\varpi} \times [\mathrm{str}\ (F_o, F_w)]^{\delta} \tag{30}$$

where (ψ, ϖ, $\delta > 0$) are parameters that are used to control the luminance, contrast, and structural components, respectively.

$$\mathrm{lum}\ (F_o, F_w) = \frac{(2 \times \lambda \times F_o \times \lambda \times F_w) + S1}{(\lambda \times F_o)^2 + (\lambda \times F_w)^2 + S1} \tag{31}$$

$$\mathrm{con}\ (F_o, F_w) = \frac{(2 \times \eta \times F_o \times \eta \times F_w) + S2}{(\eta \times F_o)^2 + (\eta \times F_w)^2 + S2} \tag{32}$$

$$\text{str}\,(F_o,\ F_w) = \frac{(2\times\eta\times F_o\times F_w)+S3}{(2\times\eta\times F_o\times\eta\times F_w)+S3} \quad (33)$$

The value of SSIM $\in [1, 0]$ and the maximum value of 1 corresponds to the optimal QoV.

B. Watermark Similarity Metrics

Here, we use the correlation coefficient metric that was proposed by Lee et al. [37]. This metric is widely used in statistical analysis, pattern recognition, and image processing. For monochrome digital images, the correlation coefficient is defined as:

$$Corr_{Cof} = \frac{\sum_i (x_i - x_m)\times(y_i - y_m)}{\sqrt{\sum_i (x_i - x_m)^2}\times\sqrt{\sum_i (y_i - y_m)^2}} \quad (34)$$

where x_i and y_i are the intensity values of the i^{th} pixel in the original watermark (W_o) and the extracted watermark (W_{ext}), respectively. The maximum value of the correlation coefficient metric is 1, which is attained when the two watermarks are identical. When the value of the correlation coefficient metric is 0, the two watermarks are completely uncorrelated. When the value of the correlation coefficient metric is -1, the two watermarks are completely anti-correlated. In this context, we employ the correlation coefficient metric to evaluate the resistance of the proposed video marking system against the attacks that are listed in the threat model above.

C. Audio Watermarking Metrics

To evaluate the audio watermarking performance, we use the PSNR metric, where (F_o) and (F_w) are replaced by (AS_o) and (AS_w), which represent the original audio signal and the watermarked audio signal, respectively. In addition, we use the waveform difference of the audio signals (i.e., before and after watermarking) to graphically demonstrate the similarity between the original audio and the watermarked audio.

VI. Experimental Results and Evaluations

In this section, we present the results of our experiments in terms of the metrics that were described in the previous section. In addition, the results are compared with previous works that were discussed in the related work section.

A. System Setup

The proposed video watermarking system is implemented using the Matlab programming language. The system is executed on a laptop that has a Genuine Intel® 2.4 GHz PC with 4.00 G RAM and is running Microsoft Windows 7 Ultimate. We apply our proposed video watermarking system to a rhino video and use the logo of Naif Arab University for Security Sciences as a watermark, as shown in Fig. 14.

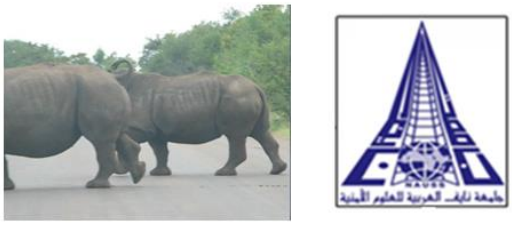


Fig. 14. Video and Watermark.

TABLE IV. Rhino Video

Name	Length	Number of Frames	Extension
Rhinos.	7 seconds.	144.	AVI.

Table 4 briefly describes this rhino video.

Our proposed watermarking system can be applied to videos that have other extension formats if they are converted into the AVI extension format.

B. Evaluations

The following table lists the works to which we compare our proposed system.

1) PSNR-based QoV evaluation: Under increased values of power factors (μ_{TC}) that conrol the transperancy, we evaluate our proposed MBD approach in comparison with the I-frames and Key-frames approaches. Fig. 15 presents the results.

Discussion. Among the approaches in Fig. 15, the MBD approach occupies the first rank, followed by the I-frames and Key-frames approaches. The reason behind the best performance of the MBD approach is that error (or noise) that is caused by hiding the watermark is minimal, as it propagates within the moving parts of the frames. By contrast, this error is centered in the I frames or other frames in the other approaches, which, in turn, deteriorates the QoV. In the Key-frames approach, the three types of frames (the I, B, and P frames that form a video) may contain the watermark if it is embedded within some motion (or some moving blocks) that is formed by the sequence of the three previous frames. This type of embedding leads to the maximization of the PSNR values compared to hiding in the I-frames only.

TABLE V. Approach Descriptions

Type	Approach	Location of Hiding	Hiding Technique
Visual Stream	[20]	Key-frames	DWT
	[23]	I-frames	DCT
Audio Stream	[26]	Original audio stream	Spread spectrum
	[27]	Original audio stream	LBS

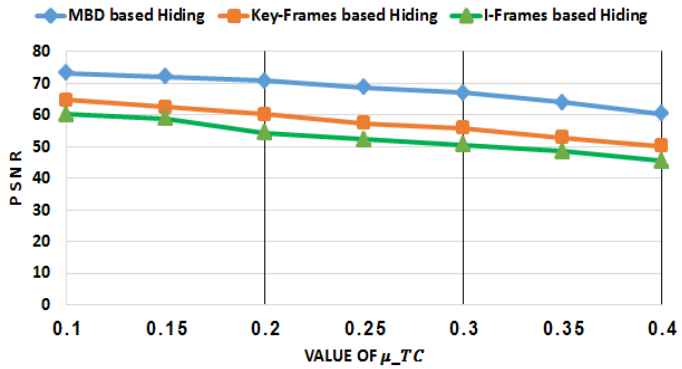


Fig. 15. μ_{TC} Values vs. PSNR.

2) SSIM-based QoV evaluation: Fig. 16 shows the results that were obtained under increased values of the power factors (μ_{TC}) that conrol the transperancy.

Discussion. The results shown in Fig. 16 support those shown in Fig. 15 because there is an inverse relationship between the QoV and the frame quality degradation. In other words, if the frame quality degradation decreases, the QoV increases, which results in higher SSIM values. The amount of quality degradation in the frames (when using the MBD-based hiding approach) is the lowest; hence, it outperforms the key-frames- and I-frames-based hiding approaches. The key-frames-based hiding approach outperforms the I-frames-based hiding approach due to the smaller amount of error caused by hiding the watermark. However, sometimes, the watermark is embedded in a key frame that includes a high moving block frequency, which explains the results that were obtained in the third and final trial (i.e., when $\mu_{TC} = 0.2, 0.4$). Therefore, under the SSIM metric, the key-frames-based hiding approach yields results that are close to those of the MBD-based hiding approach in such cases.

In evaluating the proposed video watermarking system under the attacks, we follow the following strategy: (1) the system is run (i.e., hide the watermark); (2) the attacks in the threat model are applied; (3) the extraction process is performed to obtain the watermark; and (4) the correlation coefficient metric is used to extract the results (i.e., we calculate the similarity between the original watermark and the extracted one using the correlation coefficient metric).

3) Impact of the bilinear attack: After applying the bilinear attack on the watermarked video, the extracted watermark is distorted. Fig. 17 shows the original and extracted watermarks.

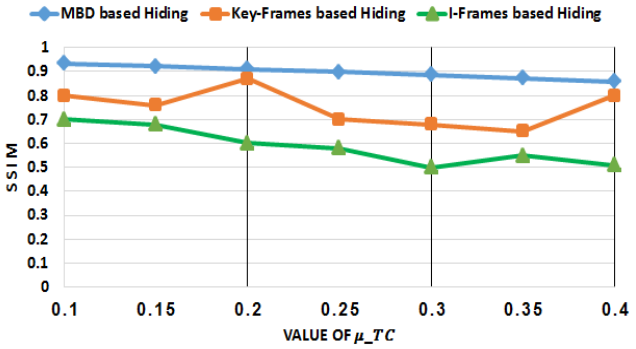


Fig. 16. μ_{TC} Values vs. SSIM.

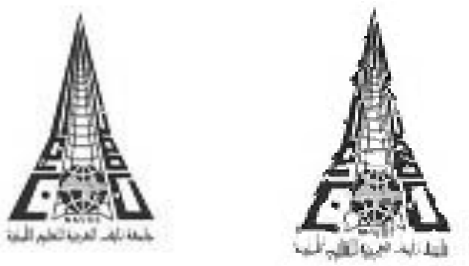


Fig. 17. Original and Extracted Watermarks after Applying the Bilinear Attack.

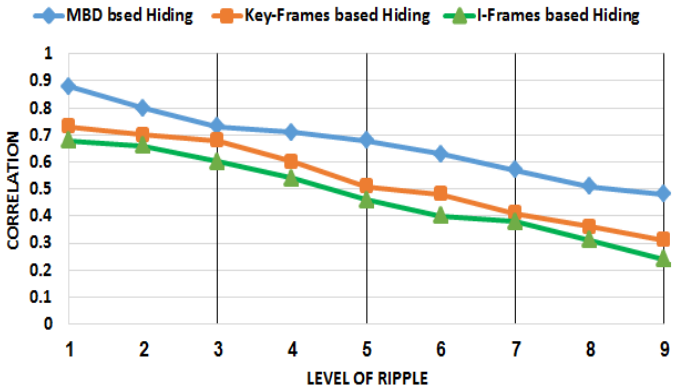


Fig. 18. Correlation Vs. the Level of Ripple under a Bilinear Attack, where $\mu_{TC} = 0.25$.

Under an increased level of ripple and power factors of ($\mu_{TC} = 0.25$), we calculate the correlation values, which are plotted in Fig. 18.

Discussion. There is an inverse relationship between the level of ripple and the correlation value. Therefore, the values of the correlation are decreased when the level of the ripple is increased in the all compared approaches. However, the proposed MBD-based hiding approach yields the best results because a high percentage of ripple levels are included in the moving blocks that are used to hid the watermark, resulting in a small effect of the bilinear attack and hence the highest similarity between the original and extracted watermarks and the highest resistance against the bilinear attack. In the key-frame-based hiding approach, the selected key frames may include many moving parts, which contain a considerable percentage of the ripple levels of the original. Hence, the approach ranks second in terms of resistance against the bilinear attack. The I-frame-based hiding approach performs the worst since none of the ripples are originally included in the I-frames that are selected for hiding the watermark. Consequently, it has the lowest resistance against the bilinear attack.

4) Impact of the curved attack: After applying the curved attack on the watermarked video, the extracted watermark is distorted. Fig. 19 shows the original and extracted watermarks.

Under an increased level of ripple and power factors of ($\mu_{TC} = 0.25$), we calculate the correlation values, which are plotted in Fig. 20.

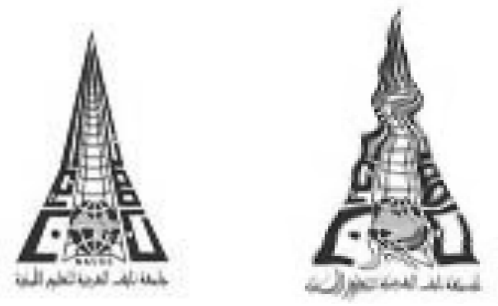


Fig. 19. Original and Extracted Watermarks after Applying the Curved Attack.

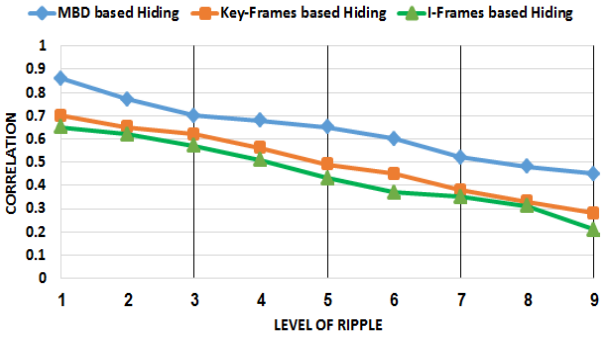


Fig. 20. Correlation vs. the Level of Ripple under the Curved Attack, with $\mu_{TC} = 0.25$.

Discussion. The curved attack can be viewed as an expanded bilinear attack because the curved attack negatively affects each part of the embedded watermark (i.e., each line that is drawn in the watermark is distorted in an arc-like manner). For this reason and due to the nature of the watermark that is used in this work (i.e., it includes many connected straight lines), the values of the correlation that are plotted in Fig. 20 are slightly lower compared to those that are plotted in Fig. 18. However, the MBD-based hiding approach still performs the best among the compared approaches against the curved attack. The same justification as was offered for the results that were obtained when applying the bilinear attack holds here.

5) Impact of the LPF attack: After applying the LBF attack on the watermarked video, the extracted watermark is distorted. Fig. 21 shows the original and extracted watermarks.

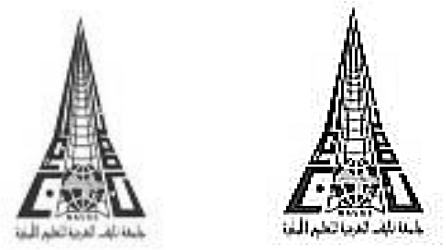


Fig. 21. Original and extracted watermarks after applying the LPF attack.

Under increased filter sizes and power factors of ($\mu_{TC} = 0.25$), we calculate the correlation values, which are plotted in Fig. 22.

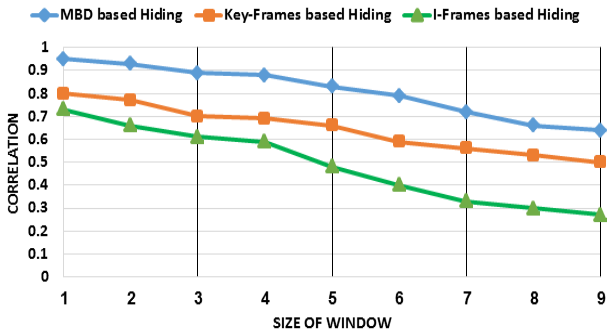


Fig. 22. Correlation Vs. the Size of the Window under the LPF Attack, where $\mu_{TC} = 0.25$.

Discussion. According to Fig. 22, the correlation value decreases as the window size of LBF increases in all three approaches. The MBD-based hiding approach performs the best under the LPF attack threat. That is because the smoothness of the moving blocks in the host frames is not affected substantially by the LPF attack, which protects the embedded watermark from degradation. The reason is that the degradation of the smoothness can be viewed as a type of blurring, which is originally included in the motion. Therefore, the original blurring of the moving blocks can disperse the blurring that is added by the LPF attack. Thus, the similarity between the original watermark and the extracted one is the highest. The I-frame-based hiding approach does not cause any blurring since no motion is created by the I-frames of a video. Hence, the host frame is substantially affected by the LPF attack, which results in a high dissimilarity between the original watermark and the extracted one. Consequently, the I-frame-based hiding approach has the lowest resistance against the LPF attack. In the Key-frame-based hiding approach, motion is formed by the key frames, which mitigates the negative impact of the LPF attack and results in moderate correlation values.

6) Impact of the Gaussian noise attack: After applying the Gaussian noise attack on the watermarked video, the extracted watermark is distorted. Fig. 23 shows the original and extracted watermarks.

Under an increased noise percentage and power factors of ($\mu_{TC} = 0.25$), we calculate the correlation values, which are plotted in Fig. 24.

Discussion. According to Fig. 24, the correlation value substantially decreased as the noise percentage increased for all three approaches due to external and new parts (i.e., the noise points or signals) being added to the original frame, which affects the resolution of each pixel of the host frame. This decrease leads to a highly distorted extracted watermark, which results in a poor correlation value. However, the MBD-based hiding approach yields correlation values that are in the range of [0.4 - 0.8] compared to [0.25 – 0.64] and [0.13 – 0.55] in the key-frame- and I-frame-based hiding approaches, respectively, which corresponds to a correlation average of 60 % in the MBD-based hiding approach, 45 % in the key-frame-based hiding approach, and 34 % in the I-frame-based hiding approach. The MBD-based hiding approach has the highest resistance against the Gaussian noise attack, which is due to selection of a suitable place for the watermark to be embedded.

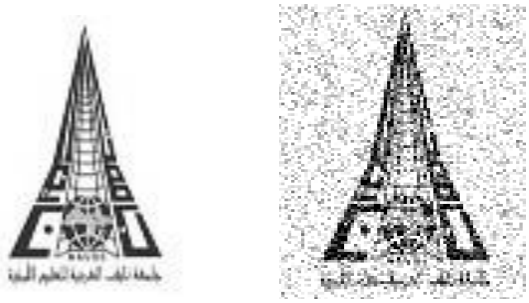


Fig. 23. Original and Extracted Watermarks after Applying the Gaussian Noise Attack.

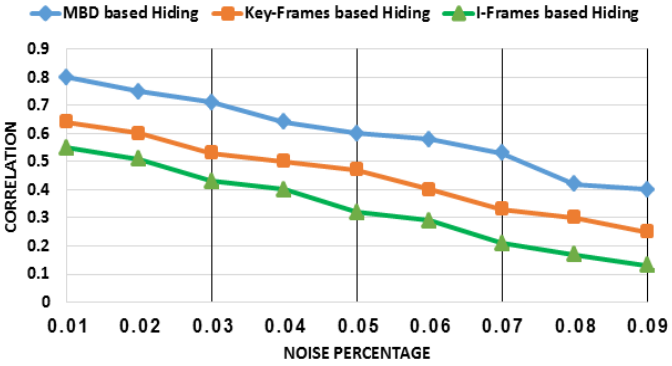


Fig. 24. Correlation Vs. Noise Percentage under the Gaussian Noise Attack, with $\mu_{TC} = 0.25$.

7) Impact of the compression attack: After applying the compression attack on the watermarked video, the extracted watermark is distorted. Fig. 25 shows the original and extracted watermarks.

Under an increased compression level and power factors of ($\mu_{TC} = 0.25$), we calculate the correlation values, which are plotted in Fig. 26.

Discussion. Compared to the Gaussian noise attack, Fig. 26 shows that under the threat of the compression attack (i.e., increasing compression level), the correlation value dramatically decreased in all three approaches, especially in the key-frame- and I-frame-based approaches, because the compression attack negatively affects both the resolution of the pixels and the contrast of the host frame and, consequently, the embedded watermark. However, the MBD-based approach preserves its resistance against the compression attack and is assigned the top ranking. The corresponding range within which the correlation value varies is [0.37 – 0.72], compared to [0.1 – 0.6] and [0.07 – 0.51] for the key-frame- and I-frame-based hiding approaches. The ranges correspond to 51 %, 35 %, and 29 % correlation averages. The reasons behind the highest resistance of the MBD-based approach are as follows: (1) it uses DWT as the hiding technique, which is resistant against the compression attack, and (2) selects the moving parts of the host frames for hiding the watermark. The key-frame-based hiding approach has a higher resistance against the compression attack than the I-frame-based hiding approach. because the key-frame-based hiding approach relies on DWT as a hiding technique, while the I-frame-based hiding approach relies on DCT as a hiding technique. DCT has a lower resistance against the compression attack compared to DWT [38].

To evaluate the proposed SDA-based audio watermarking approach, we calculate the PSNR values of the approaches that are related to the audio stream and listed in Table 5 above. Table 6 presents the results, along with the corresponding extracted watermarks.

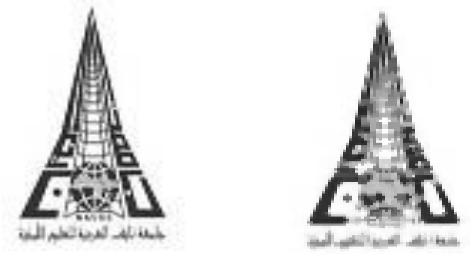


Fig. 25. Original and Extracted Watermarks after Applying the Compression Attack.

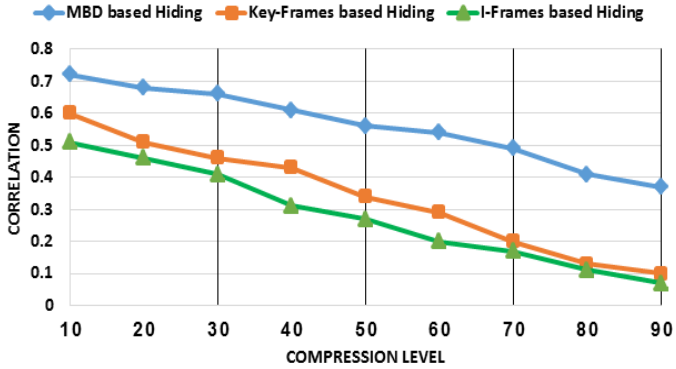


Fig. 26. Correlation Vs. Compression Level under the Compression Attack, with $\mu_{TC} = 0.25$.

TABLE VI. APPROACH DESCRIPTIONS

Approach	PSNR Value	Extracted Watermark
SDA-based hiding.	70.658	
Spread-spectrum-based hiding.	68.896	
LBS-based hiding.	60.221	

Discussion. PSNR yields lower values when watermarking audio streams compared to visual streams. That is because of the nature of the audio signals: human ears are more sensitive to changes than human eyes. However, the LBS-based hiding approach performs the worst since it depends on the spatial

domain in the hiding process. The spread-spectrum- and the SDA-based hiding approaches yield similar PSNR values since both depend on the frequency domain in the hiding process. The SDA-based hiding approach yields the highest PSNR value. The reason is that DWT is more accurate in manipulating the frequencies of the audio stream compared to the spread-spectrum-based hiding approach [38].

Regarding resistance against the Gaussian noise attack, Fig. 27 shows the waveform differences of the audio signals.

Discussion. According to Fig. 27, the proposed SDA-based hiding approach performs the best and has the highest resistance against the compression attack. That is because of the silence deletion, where the watermark is embedded within the pure (or cleaned) original audio stream. The spread-spectrum-based hiding approach does not take into consideration the silence deletion, which leads to a large difference between the original audio stream and the watermarked one. The LBS-based hiding approach performs the worst, with the largest difference between the original audio stream and the watermarked one. The reasons are as follows: (1) it depends on the spatial domain in hiding process and (2) it does not take into consideration the silence deletion, resulting in the lowest resistance against the compression attack.

Regarding the resistance to the Gaussian noise attack, Fig. 28 shows the waveform differences of the audio signals.

Discussion. The Gaussian noise attack has a stronger negative impact compared to the compression attack when they are applied on audio signals [39], which justifies the larger difference between the waveforms for all the approaches, as shown in Fig. 28. The SDA-based hiding approach has the highest resistance against the Gaussian noise attack, with the smallest difference between the original audio stream and the watermarked one. The spread-spectrum-based hiding approach is ranked second in terms of resistance against the Gaussian noise attack. The LBS-based hiding approach has the weakest resistance against the Gaussian noise attack. The silence deletion step being used in the SDA-based hiding approach but not in the other approaches plays a significant role in justifying these results.

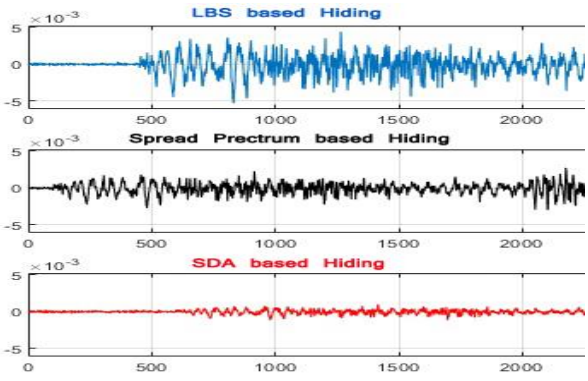


Fig. 27. Waveform differences between the Original Audio Streams and the Watermarked Audio Streams in the Three Approaches under the Compression Attack.

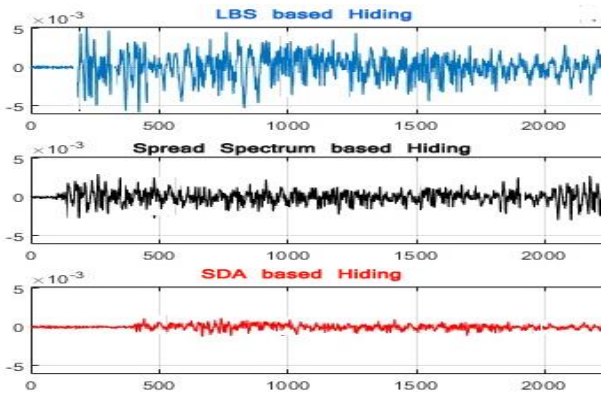


Fig. 28. Waveform differences between the Original Audio Streams and the Watermarked Audio Streams in the Three Approaches under the Gaussian Noise Attack.

VII. Conclusions

Video watermarking is a powerful method for ensuring copyright protection of digital multimedia content. The integrity of the watermarked video (in both the visual and audio streams), high quality of the watermarked video, transparency of the embedded watermark, and resistance against attacks (geometric and non-geometric) are top requirements in any video watermarking system. In this work, we propose a component-based video marking system that satisfies these requirements. The components are as follows: Recorder_{OV} , Splitter , Finder_{HP} , Hider_{DWT} , Remover_{S} , Hider_{DCT}, Adder_{S}, and Mixer. The Finder_{HP} component finds a place to hide the watermark within the visual stream. The Finder_{HP} component executes a moving block detection (MBD) algorithm to form the hiding place. The process of hiding in the visual stream is executed by the Hider_{DWT} component, which depends on DWT. Regarding watermarking the audio stream, the Hider_{DCT} component uses DCT to hide the watermark in the pure original audio stream. The pure original stream is obtained by the Remover_{S} component, which is responsible for deleting the noise from the original audio stream by executing a silence deletion algorithm (SDA). The proposed system is tested under various geometric and non-geometric attacks. According to the quality of video (QoV) metrics, namely, PSNR, SSIM, and the correlation coefficient, the proposed system is highly resistant against the attacks compared to similar systems that watermark the visual stream. Moreover, according to the PSNR and waveform difference metrics, the proposed system is highly resistant against attacks compared to similar systems that watermark the audio stream.

In future work, we will extend the proposed video watermarking system to deal with additional attacks, such as rotation and bilinear-curved attacks. In addition, we intend to satisfy the capacity requirement, which was not considered in this work.

References

[1] Kim, Hee-Dong, et al. "Robust DT-CWT watermarking for DIBR 3D images." IEEE Transactions on Broadcasting 58.4 (2012): 533-543.

[2] Bhatt, Santhoshi, et al. "Image steganography and visible watermarking using LSB extraction technique." Intelligent Systems and Control (ISCO), 2015 IEEE 9th International Conference on. IEEE, 2015.

[3] Darshan, B. R., and S. A. K. Jilani. "Digital Video Watermarking Using Discrete Cosine Transform and Perceptual Analysis." International Journal of Computer Science and Network Security (IJCSNS) 13.9 (2013): 66.

[4] Kaur, Ramanjeet, Arwinder Kaur, and Shalini Singh. "A Survey and Comparative Analysis on Video Watermarking." (2016).

[5] Chen, Yueh-Hong, and Hsiang-Cheh Huang. "Coevolutionary genetic watermarking for owner identification." Neural Computing and Applications 26.2 (2015): 291-298.

[6] Bianchi, Tiziano, and Alessandro Piva. "Secure watermarking for multimedia content protection: A review of its benefits and open issues." IEEE Signal Processing Magazine 30.2 (2013): 87-96.

[7] Qi, Xiaojun, and Xing Xin. "A singular-value-based semi-fragile watermarking scheme for image content authentication with tamper localization." Journal of Visual Communication and Image Representation 30 (2015): 312-327.

[8] Levy, Kenneth L. "Digital watermarking and fingerprinting applications for copy protection." U.S. Patent No. 9,349,411. 24 May 2016.

[9] Liu, Li, Tao Guan, and Zutao Zhang. "Broadcast monitoring protocol based on secure watermark embedding." Computers & Electrical Engineering 39.7 (2013): 2299-2305.

[10] Lubin, Jeffrey, Jeffrey A. Bloom, and Hui Cheng. "Robust content-dependent high-fidelity watermark for tracking in digital cinema." Electronic Imaging 2003. International Society for Optics and Photonics, 2003.

[11] Araghi, Tanya Koohpayeh, et al. "A Survey on Digital Image Watermarking Techniques in Spatial and Transform Domains." (2016).

[12] Buhari, Adamu Muhammad, et al. "Low complexity watermarking scheme for scalable video coding." Consumer Electronics-Taiwan (ICCE-TW), 2016 IEEE International Conference on. IEEE, 2016.

[13] Singh, Prabhishek, and R. S. Chadha. "A survey of digital watermarking techniques, applications and attacks." International Journal of Engineering and Innovative Technology (IJEIT) 2.9 (2013): 165-175.

[14] Nasir, Ibrahim, Ying Weng, and Jianmin Jiang. "A new robust watermarking scheme for color image in spatial domain." Signal-Image Technologies and Internet-Based System, 2007. SITIS'07. Third International IEEE Conference on. IEEE, 2007.

[15] AL-RAHAL, M. SHADY, A. D. N. A. N. ABI SEN, and ABDULLAH AHMAD BASUHIL. "HIGH LEVEL SECURITY BASED STEGANORAPHY IN IMAGE AND AUDIO FILES." Journal of Theoretical and Applied Information Technology 87.1 (2016).

[16] Jiang Xuehua,―Digital Watermarking and Its Application in Image Copyright Protection‖, 2010 International Conference on Intelligent Computation Technology and Automation.

[17] Malvar, Henrique S., and Dinei AF Florêncio. "Improved spread spectrum: a new modulation technique for robust watermarking." IEEE transactions on signal processing 51.4 (2003): 898-905.

[18] Potdar, Vidyasagar M., Song Han, and Elizabeth Chang. "A survey of digital image watermarking techniques." INDIN'05. 2005 3rd IEEE International Conference on Industrial Informatics, 2005.. IEEE, 2005.

[19] Kim, Dug-Ryung, and Sung-Han Park. "A robust video watermarking method." Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on. Vol. 2. IEEE, 2000.

[20] Ahmed A. Baha'a Al-Deen, Abdul Rahman Ramli, Mohammad Hamiruce Marhaban, ,Syamsiah Mashohor, "Improving Invisibility of Blind Video Watermarking Scheme", The 5th Student Conference on Research and Development -SCOReD 2007, 11-12 December 2007, Malaysia.

[21] Vivek Kumar Agrawal on "Perceptual Watermarking Of Digital Video Using Variable Temporal Length 3D- DCT ", Thesis, Department of Electrical Engineering, IIT Kanpur, 2007.

[22] Agarwal, Vivek, and Sumana Gupta. "Variable temporal length 3D-DCT based video watermarking." 20th Annual IS&T/SPIE Symposium on Electronic Imaging. 2008.

[23] Masoumi, Majid, and Shervin Amiri. "Copyright Protection of Color Video Using Digital Watermarking." International Journal of Computer Science Issues (IJCSI) 9.4 (2012): 91.

[24] Masoumi, Majid, and Shervin Amiri. "A high capacity digital watermarking scheme for copyright protection of video data based on YCbCr color channels invariant to geometric and non-geometric attacks." International Journal of Computer Applications 51.13 (2012).

[25] Petrovic, Rade, and Dai Tracy Yang. "Audio watermarking in compressed domain." Telecommunication in Modern Satellite, Cable, and Broadcasting Services, 2009. TELSIKS'09. 9th International Conference on. IEEE, 2009.

[26] Shokri, Shervin, Mahamod Ismail, and Nasharuddin Zainal. "Voice quality in speech watermarking using spread spectrum technique." Computer and Communication Engineering (ICCCE), 2012 International Conference on. IEEE, 2012.

[27] Cvejic, Nedeljko, and Tapio Seppanen. "Increasing the capacity of LSB-based audio steganography." Multimedia Signal Processing, 2002 IEEE Workshop on. IEEE, 2002.

[28] Fallahpour, Mehdi, and David Megías. "Audio watermarking based on Fibonacci numbers." IEEE Transactions on Audio, Speech, and Language Processing 23.8 (2015): 1273-1282.

[29] Qiang Cheng, Thomas S. Huang, Hao Pan, "COMBINED AUDIO AND VIDEOWATERMARKING USING MEL-FREQUENCY CEPSTRA", IEEE International Conference on Multimedia and Expo, ISBN 0-7695-1198-8/01 $17.00 2001 IEEE.

[30] Dittmann, Jana, and Martin Steinebach. "Joint watermarking of audio-visual data." Multimedia Signal Processing, 2001 IEEE Fourth Workshop on. IEEE, 2001.

[31] Fung, Yik-Hing, and Yuk-Hee Chan. "Green noise video halftoning." Digital Signal Processing (DSP), 2014 19th International Conference on. IEEE, 2014.

[32] Zoom website, (2018) online available: https://zoom.us/.

[33] Filmora website, (2018) online available: https://filmora.wondershare.com/video-editing-tips/separate-audio-from-video.html.

[34] Anjum, Shaikh Rakhshan, and Priyanka Verma. "Performance evaluation of DWT based image watermarking using error correcting codes." Int. J. Adv. Comput. Res 2 (2012): 151-156.

[35] Dejey, D., and R. S. Rajesh. "Robust discrete wavelet-fan beam transforms-based colour image watermarking." IET Image Processing 5.4 (2011): 315-322.

[36] Yu, Dong, and Li Deng. AUTOMATIC SPEECH RECOGNITION. SPRINGER LONDON Limited, 2016.

[37] Lee Rodgers, Joseph, and W. Alan Nicewander. "Thirteen ways to look at the correlation coefficient." The American Statistician 42.1 (1988): 59-66.

[38] Saleh, H. I., et al. "Comparisons Of DCT-Based And DWT-Based Watermarking Techniques." Proc. Int. J. Sci. Res.. Vol. 16. 2006.

[39] Golestani, Hossein Bakhshi, and Shahrokh Ghaemmaghami. "Enhance Robustness of Image-in-Image Watermarking through Data Partitioning." arXiv preprint arXiv:1501.01758(2015).