Collusion-Resistant Digital Video Watermarking For Copyright Protection

Application

Azizah Abdul Manaf1, Ali Boroujerdzade1 and Seyed Mojtaba Mousavi*1

1Advanced Informatics School (AIS), Menara Razak, Universiti Teknologi Malaysia (UTM), Jalan Sultan Yahya Petra, 54100 Kuala Lumpur, Malaysia.

Abstract

In this paper, a frame-by-frame blind video watermarking technique, which is resistant against collusion attack, is proposed. Collusion attack is the main problem in video watermarking for copyright protection. Whereby in this attack, colluders try to estimate and remove the watermark by comparing with as many as watermarked materials. To address the collusion attack in this work, each frame is assigned a different watermark. Since each frame has a unique watermark, it is impossible for the attack to estimate it. To achieve the required robustness as well as maintaining the quality, a combination of Block Truncation Coding (BTC), Discrete Wavelet Transform (DWT) and Singular Value Decomposition (SVD) techniques is used. The results show that the proposed method is robust against all types of collusion attack, and at the same time, the quality of the video is maintained at a satisfactory level.

Keywords: video watermarking, collusion attack, copyright protection

Introduction

The multimedia technology and the way of using this technology has grown and changed significantly during the past decade. Therefore, legal distribution of these materials in a secure way has become a necessity. Digital works in image, audio and video formats are becoming the best way of transmitting the content, especially using the internet as the carrying platform. In the digital world, due to the ease of creating, publishing and tampering of works, copyright protection became a major issue[1, 2]. This issue is much significant in movie industry which cause millions of dollars losses[3]. With new technologies, almost everyone is able to produce, edit and share videos. Hence, there should be a mechanism to prove the ownership and making sure of integrity. Watermarking is one of the effective ways to help copyright protection [4-6].

Watermarking is able to assist ownership identification and approval, in addition to copy control and modify prevention. In other words, the digital watermarking is arranged to provide some prohibition methods against illegitimate duplication or abuse of digital contents [7]. The watermarking refers to the process of producing and embedding of some data into the digital media such as image or video [1]. A typical watermarking system has three main phases: generating, embedding and retrieving the watermark [8]. Generating the watermark is a very important step and depends on the application of watermarking an appropriate watermark should be generated. The embedding step is hiring an algorithm, which hides the watermark into the carrier. Retrieving step is to reverse the embedding and extract the watermark. The retrieved watermark can be verified by comparing with the trusted third party’s stored copy.

The main issue in watermarking with the purpose of copyright protection is the security that consists of different aspects. Robust watermarking, as one aspect of security, creates a channel to communicate or store information inside the carrier. In order to copy control, this channel should be secure. In other words, in robust watermarking the aim is preventing unauthorized user from detecting, adding, modifying or removing the watermark [9]. Moreover, the security is not limited to robustness only. Capacity, imperceptibility and prevailing design issues are also other parts of the security which has to be addressed [10].

Although video watermarking is based on the same techniques, applied on images, it has its own issues[11]. As videos usually have a bigger size in comparison with other media and they have inherent redundancy information, which is repeated between sequential frames, removal attacks still remains a big problem in video watermarking [12]. Among these removal attacks, collusion attack is the trading one [13-16].

Collusion attack is one of the major threats for copyright. In this attack, one or more colluders try to collect as many as possible watermarked material. Then by comparing these materials, it is possible to estimate the watermark. If the estimation is accurate enough a watermark removal can be performed. This attack has a very close relation with the number of watermarked materials that are collected by colluders. By increasing the number of collected material, the possibility of the attack is also increased. Therefore, addressing the collusion attack by providing more robust watermarking methods can improve the copyright protection.

The aim of this work is to design a blind watermarking method in the way that the proposed method be able to improve copyright protection by enhancing the robustness against collusion attack. For this purpose, a unique watermark is generated for each of the video frames. In this effort, a combination of DWT, SVD and BTC methods are used. The DWT and SVD are used to prepare the frames for embedding the watermark bits and the BTC [17, 18]is used for watermark generation. There are two steps of evaluation in this work. First step is to evaluate the quality of watermarked video by using Peak Signal-to-Noise Ratio (PSNR). The second one is error checking that shows how many bits of the watermark are
not extractable after the attack. Bit Error Rate (BER) measurement is chosen for this matter. The rest of this study is organized as follows: First, different types of collusion attack are described and related works are highlighted. Then, the main algorithm is presented. Afterwards, results and analysis are discussed. Finally, conclusion and future work of this paper are summarized in the last part.

Collusion Attacks

One of the security issues in video watermarking is collusion attack [19]. Collusion is a problem that has been pointed out for still images some time ago. It refers to a set of malicious users who merge their knowledge, i.e. different watermarked data, in order to produce illegal content, i.e. un-watermarked data. Such collusion is successful in two different distinct cases [20]. In the case of video, there are two models of collusion attack. The explanation of each type of collusion attack is as follow:

**Type I:**
In this scenario, the colluders assume that the frames of the video are not changing, for instance, the movie scene remains still for a while. However, the watermark on each frame is different from the other one. The scenario is visualized in Figure 1.

![Figure 1: Same frame with different watermarks (collusion attack of Type I)](image)

**Type II:**
In this scenario, the colluders assume that the frames of the video are changing. However, there is only one watermark, which is embedded into each frame. This scenario is visualized in Figure 2.

![Figure 2: Different frames with same watermark (collusion attack of Type II)](image)

A new scenario can be proposed to increase the robustness against the two types of collusion attack that are mentioned above. In this new scenario, frames and watermarks are changing despite of the two previous ones, which in either the frames or watermark was remaining unchanged (Figure 3).

![Figure 3: Different frames with different watermarks](image)

In this scenario, neither the similar part nor the different part is the watermark. Therefore, the attack estimation of the watermark is not even close to the real watermark. The proposed method in this work is based on this new scenario.

Related Works

In the past few years, many researchers proposed numerous video watermarking techniques. Video watermarking is an interesting field because of two main reasons: First, because the availability is significantly higher and then almost all video formats have enough redundant information that can be altered without ruining the frames. Video watermarking methods are proposed both in the spatial [21, 22] and frequency domain [23]. Compare to spatial based methods, the transform based methods such as Discrete Cosine Transform (DCT) [15, 24], DWT [25, 26], or Discrete Fourier Transform (DFT) [27], show better robustness against different attacks. However, this paper deals specifically with collusion attacks. Therefore, in the following a short literature about the previous collusion resistant methods are given; Kamkar et al. [15] proposed a collusion resistant method, which is a based on DCT. This method is also robust against the rotation attack. To make the scheme robust against the collusion attack, the embedding blocks of successive video frames are changed. For this purpose, a Pseudo Random Number (PRN) generator and a permutation vector are used. In addition, using rotation invariance property of the Complex Zernike moments makes this scheme robust against the rotation attack. For embedding the watermark bits, the square blocks, placed on the middle position of every luminance channel are used. The scheme also shows good robustness against conventional video attacks including a Rayleigh fading wireless channel.

The method proposed by [28] is resistant against collusion attack. In this paper, a logo is hidden into a colored image for fingerprinting purpose. This method is based on the averaging of middle band coefficients of DCT on an image. In comparison to conventional middle-band exchange schemes, this method shows the better robustness against various types of linear and non-linear collusion attacks.

Another collusion resistant is presented using selected sets of the frames [29]. For this purpose, frames are divided into separate sets and each set carry the same watermark. This means all frames are watermarked but only a small portion of the frames that are in the same set receive same watermark. Hence, different watermark in different frame make the method collusion resistant.

The method proposed by [30] is an event base decomposition scheme for watermarking uncompressed short video sequences. The video signal is assumed a sequence of overlapping visual components-called events. First, the matrix of spectral parameters is obtained through a Multiresolution SVD (MR-SVD) analysis for each video sequence. Then by applying a block based temporal decomposition (TD) method on this matrix, an orthogonal matrix of target vectors and an interpolating matrix of event functions are extracted. Finally, by means of a weighted sum procedure the watermark is embedded into the orthogonal target vectors matrix. A non-
blind extraction algorithm is used for extracting the embedded watermark. Lu et al. [25] specifically worked on linear collusion attack. This paper proposed a robust digital image-watermarking scheme based on subsampling and DWT. Subsampling is firstly used to construct a sub-image sequence as a video segment. Then, a random watermark sequence satisfied with Gaussian distribution is blocked-wised embedded into the DWT domain of these sub-images repeatedly using the video watermarking technique. Moreover, a watermark is detected through computing correlation between the watermark and watermarked frames. The experiment results demonstrate that the proposed scheme achieves good robustness against JPEG compression, common image processing operation and geometric distortions. Furthermore, the proposed watermarking scheme is also robust against linear collusion and other video watermarking attacks. Koubaa et al. [31] presented a new video watermarking which resists to collusion, MPEG4 compression and frame dropping attacks. This scheme is based on video mosaicking. The contribution of this paper is applying the mosaicking technique. In fact, mosaicking allows selecting an interesting area where the mark should be embedded. The idea is to insert the same mark into the same pixels, which represent the same physical point. This point is located in different places in different frames. Therefore, different frames get different watermark and the method can resist the collusion attack. The other collusion resistant method is proposed by [32]. This method is based on multiband (M-band) wavelets, genetic algorithms (GA) and fuzzy logic. First, M-band decomposition of the host image is obtained by means of wavelet transform. Then the GA algorithm is used to find threshold values for the selection of host coefficients as well as respective embedding strengths. In the extraction part, in order to increase the watermark decoding performance multiple stage detection through cancelation of multiple bit interference (MBI) effect is used, which is implemented by fuzzy logic algorithm.

As can be seen in this literature, different solutions are proposed to address the collusion problem. All these solutions can be categorized in two groups. In the first group, the solutions are based on embedding algorithms. In other words, the focus in this class is on the embedding process to increase the robustness against the collusion attack. The second category is to hire some secure codes in order to make the attack impossible. The restriction of these two groups is that, they cannot resist against different collusion methods. As mentioned in Section 2, two types of collusion attack are exist. First, the attacker has access to many different watermarked video frames with the same watermark. Second, the colluder has access to one video frame with different watermarks. Subsequently, by changing the watermark for each video frame the collusion attack become impossible. The proposed method in this work is based on this solution.

Proposed Scheme
Proposed Framework Overview
The general overview of the proposed framework is illustrated in context diagram (Figure 4). As can be seen in this diagram, there are two primary and two secondary inputs as well as one primary and three secondary outputs. Video file and watermark file are primary inputs of the system, which are given to the system when the embedding process is requested. The output of this request is the primary output of the system, watermarked video. Then the watermarked video is given to the system as a secondary input to request PSNR checking. The PSNR is one of the secondary outputs. Then again, the collusion attack is simulated in watermarked video and it is given to the system along with extracting request. The output of this request is extracted watermark, which is another secondary output. After attacking the extracted watermark, it is given to the system to check the BER. The BER value is the last output.

![Figure 4: Context diagram](image)

Level 0 Diagram
In this level, the general system from context diagram is broken down (Figure 5). The system is built from a combination of four subsystems. The first subsystem (0) is embedding which is responsible for the embedding request. It gets the video file and watermark file for embedding purpose. Next sub-system (2) is a quality check system. It gets the video file and watermarked video for embedding purpose. Next sub-system (2) is a quality check system. It gets the video file and watermarked video to calculate the PSNR. The extracting sub-system (1) is coming next. This subsystem gets the watermarked video and extracts the watermark from it. The extracted watermark is given to last sub-system (3), error check, to calculate the BER.
Level 1 Diagram
This level is the most detail one. It is still based on the two previous diagrams. This diagram is detailed enough to be used as the foundation for implementation (Figure 6).

Implementation of Embedding
Embedding process in this work, consists of four main parts; watermark preparation, video preparation, embedding bits and video regeneration. Detail explanation of each part is as follow.

Watermark Preparation
To address collusion attack the best way is to watermark each frame with a different watermark. To do so a unique watermark should be generated for each frame. To ease the process and prevent too much complication, only one watermark is going to be embedded. However, to have different watermarks for each frame the original watermark will be rotated for each frame. Three following steps are used for watermark preparation before going through the embedding process.

Step 1: Extracting Features from Frame
The rotation must be random, so there should be a unique angle for each frame. To have such a random angle, the best way is to use the frame itself. In this work, all pixel values of the frame added up to generate a unique number for each frame and then using modular arithmetic to generate an angle for each frame.

Step 2: Block Truncation Coding
There are two types of image compression, lossy and lossless. Block Truncation Coding or BTC categorized as a lossy one. This technique is settled for grayscale images. The method is to keep the mean and standard deviation unchanged while the levels of gray are reduced. To do so, each block of the image is a quantized. Since our aim of this work is copyright protecting, despite the cover, the quality of the watermark is irresolution. Compressing the watermark improves the PSNR of the method. So the output of the BTC process is considered as watermark in this work instead of the original watermark [17, 33]. The pseudo code for BTC is shown in Figure 7.

Step 3: Rotation
The processed watermark from BTC method is rotated according to the extracted features from each frame. This makes a unique watermark for each frame. One of the complexities in the proposed model is that the implementation becomes time and computation power consuming. Since it is required to have a unique image for each frame and embedding into all frames this complexity is happening. However, as the rotation is rather a simple process, it is a

Figure 6: Level 1 diagram
On the left side, all steps of embedding are breaking down. When the video file and watermark file are given to the system, it starts with preparing the video in 0.1. The frames will be given to 0.2 for separating the RGB matrices. In 0.3 and 0.4, DWT and SVD are applied on the matrices. At the same time, the watermark file is prepared in 0.0. Then in 0.5, the watermark’s bits are embedded into the video. The 0.6 to 0.8 are reversed previous processes to regenerate the video. On the second column, the extracting is happening. The watermarked video is given to the system in 1.0. dived it into RGB. Steps 1.1 and 1.2 are DWT and SVD that are applied on the matrices. In 1.3, the bits of the watermark are extracted from the file. Yet there are two other steps. Step 2.0 is responsible for quality checking. In this step, the quality is measured by PSNR. Moreover, in 3.0 the errors of the system are measured by the BER. The watermarked video is given into 3.0 after the collusion attack.

Figure 7: Pseudo code for Block Truncation Coding

```
Read image
Divide image into blocks (4x4)
For each block
  Calculate mean
  If pixel value is greater than the mean
    Replace it with 1
  Else
    Replace it with 0
End
```

End
good method to use only one watermark and rotates it in order to produce a unique watermark for each frame instead of preparing different images as watermarks. Furthermore, the rotation itself is a geometric attack; it means the watermarked work is attacked intentionally. This makes a great level of confusion in the process and the outcome will be more robust.

Video Preparation
The following five steps are used for video preparation before going through the embedding process.

Step 1: Separating Audio Signal from Video
First, it needs to separate the audio signal from the sequence of the frames in order to protect the audio signal from any unintentional changes. To separate two signals from each other a shell-based tool, called “ffmpeg” is used. “dos” command in Matlab makes it possible to call a shell-based tool from Matlab. The audio signal will be saved as an mp3 formatted file to reuse in regenerating the video.

Step 2: Dividing the Video into its Frames
Because of the idea to have different watermarks for each frame, it is required to divide the video signal into its frames. Each frame is separated first and then is saved as a single image.

Step 3: RGB Color Model
In the meantime, most of the Matlab functions are prepared for grayscale images and most of the watermarking works are in grayscale because of this. Nevertheless, in this work the quality of the cover is the key point so making it grayscale is unacceptable. To overcome this issue the RGB color model is hired[34]. To reproduce a comprehensive collection of colors, the RGB color model uses the primary colors, which are red, green and blue. The name of the model is the initiative of these colors. As an additive one, this model combines different portions of each color to represent a vast variety of non-primary colors. The RGB model is proposed to demonstrate imageries in electronic systems like computers. The philosophy behind it is established on human perception of colors. As the first step of embedding process, the image is divided into its colors and then the watermark is embedded into each color unconnectedly. The results of the experiments show that the embedding into colors independently improves both quality and noiselessness.

Step 4: Discrete Wavelet Transform (DWT)
DWT watermarking methods are robust against the attacks such as compression, averaging and frame dropping. Moreover, DWT based watermarking method can be perceptually invisible and blind retrieval is also possible [35]. By using DWT, an image can be split into four bands based on approximation; lower resolution (LL), horizontal (HL), vertical (LH) and diagonal (HH). This process can be repeated as many scales as needed. Among the other transforms, such as FFT and DCT, it is proved that the DWT is more accurate based on HVS. Therefore, in locations that the HVS is less sensitive watermark with higher energy can be embedded. The sensitivity of the HVS is minimum in high resolution bands, so embedding in these areas provide a good level of robustness with least impact on quality[36]. In this work, a single level of DWT with “Haar” mode is used, which has the best result in terms of quality.

Step 5: Singular Value Decomposition (SVD)
An image is represented as a matrix in Matlab, which can be a real, or a complex one. SVD is a numerical analyzing technique, which helps to prepare the frames in the video to embed the watermark. By using this technic the retrieving process can be blind and there is no need to have the original host [37]. The concept SVD, goes back to the linear algebra theorem [38]. It shows that a rectangular matrix can be decomposed into three matrices (with non-negative real entries) $U$, $V$ and $V^T$ are orthogonal square matrices and $S$ is a rectangular diagonal matrix with its values arranged in descending order. Mathematical representation of a square matrix $A$ of order of $N$ after SVD transform is given as follows [39-42]:

$$A = U S V^T = \begin{bmatrix} u_1, & u_2, & ..., & u_N \end{bmatrix} \begin{bmatrix} \delta_1 & \vdots & \delta_N \end{bmatrix} \begin{bmatrix} v_1^T \vdots v_N^T \end{bmatrix}$$

where $U \in \mathbb{R}^{N \times N}$ and $V \in \mathbb{R}^{N \times N}$ are unitary matrices, i. e. $U U^T = I_N$ and $V V^T = I_N$. The columns of matrices $U$ and $V$ are called the left and right singular vectors of matrix $A$ respectively. The operand $\delta$ is used for the conjugate transpose operation. As mentioned before, $S \in \mathbb{R}^{N \times N}$ is a diagonal matrix ($S = \text{diag} (\delta_1, \delta_2, ..., \delta_N)$) with singular values $\delta_i (i = 1, 2, ..., N)$. A matrix is known as a diagonal if all entries beside the main diagonal ($\gamma$) of a square matrix are all zeros. This matrix is chosen as a location for embedding the watermark. The SVD is a built-in function in Matlab.

Embedding (Bitset)
In this step, the bits of the prepared watermark for each frame are embedded into the corresponding frame using “bitset” command in Matlab. The location for embedding the bits is already identified using previous steps.

Video Regeneration
After watermarking all frames, it is required to put back all frames and add the audio signal to the file to have the watermarked video file. To do so, there are four main steps. First is to reverse the SVD that simply can be achieved by multiplying three matrices $U$, $V$ and $S$. Second step is to inverse the DWT. The inverse DWT, gets all lower resolution (LL), horizontal (HL), vertical (LH) and diagonal (HH) bands and put them together using the same mode that is used in the DWT process. Therefore, in inverse process the mode will be “Haar” as well. Afterwards, the colors of the frame should be combined again. Finally, the audio signal is added to the sequence of watermarked frames.

Implementation of Quality Checking
Signal to Noise Ratio (SNR) is one of the methods to measure the quality of the watermarked image by comparing the watermarked image with the original one. The difference between them is represented as errors that indicates how much the watermarking changed the image. Equation (2) represents the error function.

$$e(x, y) = I(x, y) - I_w(x, y)$$

3488
Where \( I \) represents the original image and \( I_w \) is the watermarked image. By increasing the value of the error, the watermarking or attack, cause a greater distortion. The simplest way to measure this distortion, is the Mean Square Error (MSE).

\[
MSE = \frac{1}{MN} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (I(i,j) - I_w(i,j))^2
\]

(3)

where the image dimensions are shown by \( N \times M \). Using MSE, another quality measurement can be calculated. The PSNR calculates the ratio between the maximum possible energy of a signal and the energy of corrupting noise.

\[
PSNR(I, I_w) = 20 \times log_{10} \left( \frac{MAX_I}{MSE} \right)
\]

(4)

where \( MAX_I \) is the maximum possible pixel value of the original image. Larger PSNR means the original and watermarked image are more similar to each other.

Implementation of Collusion Attacks

To make sure that the proposed method in this work is robust against all types of collusion attack, the attack in both scenarios is simulated.

First Scenario:
The attacker in this scenario is looking for the difference between the frames. Since the frames are same as each other, they do not have any difference. However, when a watermark is added to the frame, it makes a difference. Therefore, if the attacker manages to find the exact difference between all frames and if the number of frames is high enough, eventually the position of the watermark and the watermark itself can be extracted. The implementation of this scenario is by using the built in “diff” function in Matlab. Each two frames are compared together and the different part will be kept. This process goes on until every possible pair of frames is compared. Then same process is applied to the kept differences. The eventual remaining should be the watermark based on the attack definition.

Second Scenario:
In this scenario, the attacker is looking for the similarity between the frames. Since the frames are different from each other, there is no similar part between them. However, when the same watermark is added to all frames, that watermark becomes the similar part between frames. Consequently, if the attacker manages to find the exact similarity between all frames and if the number of frames is high enough, eventually the position of the watermark and the watermark itself can be extracted. To implement this scenario the averaging is used. The average of each two frames is computed by adding corresponding pixel values of each frame to the other one and dividing it by 2. This process goes on until every possible pair of frames is averaged. Then same process is applied on the kept averages. The eventual remaining should be the watermark based on the attack definition.

Implementation of Extraction

In extraction part, the video file has to be disassembled again. In each frame, the watermarked is embedded in the S matrix of SVD of the HH level of the DWT of each color bands (Figure 8). The embedded watermark bits are extracted from each frame using “bitget” command in Matlab. The implementation of RGB separator, DWT and SVD is entirely same as a corresponding step in embedding.

![Figure 8: Implementation of watermark extraction](image)

**Implementation of Error Checking**

BER is defined in a digital transmission area as the number of bits that are received incorrectly during a transmission. These bits are altered due to a noise (i.e. attacks) or due to unintentional process. Nevertheless, if the embedding and extracting considered as a transmission, the BER can measure the impact of the attack in watermarking area. The following pseudo code shows the steps of BER calculation (Figure 9):

```
Read watermark
Read extracted watermark (after the attack)
Compare watermark and extracted watermark bit by bit
If the bits are not equal
Add one to BER
End
Divide BER to the length of the image
End
```

![Figure 9: Pseudo code for implementation of error checking](image)

**Results and Analysis**

**Experiment System and Materials**

Implementation of this work is in Matlab, which is one of the strongest tools in image processing to implement the design. To do the experiments a computer with following specifications is used:

<table>
<thead>
<tr>
<th>Specification</th>
<th>Specification Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Microsoft Windows 8 Pro (64-bit)</td>
</tr>
<tr>
<td>Matlab</td>
<td>R2012a (7.14.0.739) 64-bit</td>
</tr>
<tr>
<td>CPU</td>
<td>Intel Core 2 Quad Q8400S @ 2.66GHz</td>
</tr>
<tr>
<td>RAM</td>
<td>16 GB Single-Channel DDR3 @ 531MHz</td>
</tr>
<tr>
<td>Motherboard</td>
<td>Hewlett-Packard 3648h (XU1 PROCESSOR)</td>
</tr>
<tr>
<td>Graphics</td>
<td>Intel Q45/Q43 Express Chipset</td>
</tr>
</tbody>
</table>

For the experiments, the 100*100-pixel image, which is shown in Figure 10 (a), is chosen as the original watermark. The Figure 10 (b) is the same watermark after applying BTC.
In this work, 15 short videos are chosen for the experiment. Most of the videos are chosen from real movies in order to make the experiment condition as similar as possible to the real situation. In addition, the experiments is only limited to video files H. 264 format. The videos will be recall based on the numbers that is assigned to each one. Table 2 shows a sample frame from each video and the number, which is assigned to it.

<table>
<thead>
<tr>
<th>Video Number</th>
<th>Length (seconds)</th>
<th>Frame size</th>
<th>Frame rate (per second)</th>
<th>Number of frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video 1</td>
<td>10</td>
<td>512x288</td>
<td>30</td>
<td>320</td>
</tr>
<tr>
<td>Video 2</td>
<td>8</td>
<td>584x480</td>
<td>29</td>
<td>252</td>
</tr>
<tr>
<td>Video 3</td>
<td>12</td>
<td>584x480</td>
<td>29</td>
<td>366</td>
</tr>
<tr>
<td>Video 4</td>
<td>6</td>
<td>1280x720</td>
<td>29</td>
<td>198</td>
</tr>
<tr>
<td>Video 5</td>
<td>9</td>
<td>1280x720</td>
<td>29</td>
<td>291</td>
</tr>
<tr>
<td>Video 6</td>
<td>9</td>
<td>1280x720</td>
<td>29</td>
<td>294</td>
</tr>
<tr>
<td>Video 7</td>
<td>10</td>
<td>1280x720</td>
<td>29</td>
<td>323</td>
</tr>
<tr>
<td>Video 8</td>
<td>8</td>
<td>1280x720</td>
<td>29</td>
<td>258</td>
</tr>
<tr>
<td>Video 9</td>
<td>20</td>
<td>1280x720</td>
<td>29</td>
<td>600</td>
</tr>
<tr>
<td>Video 10</td>
<td>15</td>
<td>1280x720</td>
<td>29</td>
<td>453</td>
</tr>
<tr>
<td>Video 11</td>
<td>13</td>
<td>1280x720</td>
<td>29</td>
<td>390</td>
</tr>
<tr>
<td>Video 12</td>
<td>13</td>
<td>1280x720</td>
<td>29</td>
<td>392</td>
</tr>
<tr>
<td>Video 13</td>
<td>14</td>
<td>1280x720</td>
<td>29</td>
<td>440</td>
</tr>
<tr>
<td>Video 14</td>
<td>12</td>
<td>1280x720</td>
<td>29</td>
<td>367</td>
</tr>
<tr>
<td>Video 15</td>
<td>9</td>
<td>1280x720</td>
<td>29</td>
<td>275</td>
</tr>
</tbody>
</table>

Embedding Results
As it explained, 15 videos are chosen to experiment. All these videos alongside to the watermark are given to the proposed system to test the system. Videos number 4 is chosen as a sample for graphical illustration of the results in each step.

Watermark Preparation Result
The first step of watermark preparation is applying BTC on it. The result of this process is shown in Figure 10 (b). The next step is to rotate it based on the information from each frame. The rotated watermark corresponding to each presented sample frame is shown in Figure 11.

Video Preparation Results
Since the aim is frame-by-frame video watermarking it is required to have access to each frame separately. To do so the video is divided to its frames. Figure 12 shows the frame sample of video number 4.

Embedding Results
During the embedding process, there are four steps. First step is to dividing each frame into its color bands. Then DWT and SVD are applied on the frame. Finally, the watermark bits are embedded into the frame.

Step 1: RGB Bands Results
In this step, each frame is divided in to its color bands. This deviation make it possible to work with video in color more easily. In addition, embedding the watermark into different color band provide more confusion and invisibility. Each color band in Matlab is presented as a matrix. This intermediate result is illustrated in Figure 13.
Step 2: DWT Results
In proposed method, one level of DWT is applied on each color band. Figure 14 shows the output of applying DWT on each frame. As can be seen in this figure, the output of DWT has four bands. It is possible to embed the watermark to each or all of these bands. However, initial testing in this work showed that the embedding into the HH band would have the better result in terms of the quality. Hence, the next step will use this band as the chosen location of embedding.

Step 3: SVD Results
In this step, each HH band of DWT from previous step is going through SVD process. The output of SVD is illustrated in Figure 15.

Step 4: Bitset Results
In this step, each frame is watermarked using its unique watermark. To show the results, each watermark from Figure 11 is embedded into corresponding frame in Figure 12 and the watermarked frame is presented in Figure 16.

There are three matrices as output of the SVD. The S matrix is chosen for embedding the watermark. In other words, the bit in S matrix is literary replaced with the bits of the watermark sequentially.

Quality Check Results
In this step, the quality of the watermarked video is compared to the original one. The applied benchmark for this purpose is PSNR. It is generally accepted that, if the PSNR is between 30 and 70 the quality is imperceptible enough. In other words, with PSNR more than 30, human visual system is incapable of distinguishing between original and watermarked frame. As this work is a frame by frame watermarking, the PSNR of the video is calculated as the average of the PSNR of all frames in each video sample. Table 3 displays the PSNR results:

<table>
<thead>
<tr>
<th>Video</th>
<th>Average PSNR (dB)</th>
<th>Number of frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video 1</td>
<td>48.41</td>
<td>320</td>
</tr>
<tr>
<td>Video 2</td>
<td>53.57</td>
<td>252</td>
</tr>
<tr>
<td>Video 3</td>
<td>51.08</td>
<td>366</td>
</tr>
<tr>
<td>Video 4</td>
<td>55.93</td>
<td>198</td>
</tr>
<tr>
<td>Video 5</td>
<td>51.69</td>
<td>291</td>
</tr>
<tr>
<td>Video 6</td>
<td>58.80</td>
<td>294</td>
</tr>
<tr>
<td>Video 7</td>
<td>53.53</td>
<td>323</td>
</tr>
<tr>
<td>Video 8</td>
<td>55.75</td>
<td>258</td>
</tr>
<tr>
<td>Video 9</td>
<td>60</td>
<td>600</td>
</tr>
<tr>
<td>Video 10</td>
<td>51.08</td>
<td>453</td>
</tr>
<tr>
<td>Video 11</td>
<td>72</td>
<td>390</td>
</tr>
<tr>
<td>Video 12</td>
<td>58.24</td>
<td>392</td>
</tr>
<tr>
<td>Video 13</td>
<td>57.43</td>
<td>440</td>
</tr>
<tr>
<td>Video 14</td>
<td>56.43</td>
<td>367</td>
</tr>
<tr>
<td>Video 15</td>
<td>56.80</td>
<td>275</td>
</tr>
</tbody>
</table>
The minimum of PSNR is for video number 1, which is 10 seconds and has 320 frames. The maximum PSNR is for video number 11, which is 13 seconds and has 390 frames. However, the minimum frame number is for video number four with 198 frames and PSNR 55.93. The maximum frame number is for video number nine with 600 frames and PSNR 60. It means there is no meaningful relation between PSNR and duration of the video. Therefore, the changes in PSNR are related to the nature of the video. Since DWT is used in this work and HH band is chosen for embedding, it is possible to say that the illumination of the video is the reason for changing the PSNR.

Collusion Attack Simulation Result
As mentioned in Section 2, the embedding method in this work is based on the new scenario for collusion attack (Figure 3). In this scenario, frames and watermarks are changing despite of the two previous scenarios (Figure 1 and Figure 2). Therefore, the attack estimation of the watermark is not even close to the real watermark. In this step, each video is attacked and the result, which is an estimation of the watermark, is presented as images in Figure 17.

![Figure 17: Attackers' estimation of watermark in video number 4](image)

It is obvious that the estimation of attackers’ about the watermark is useless. With this estimation, it is impossible to remove the watermark. This miss estimation happens since the attack is defined based on some assumptions. The attacker assumes either the frames or the watermark is remaining unchanged during the video. Nevertheless, by changing the watermark alongside with dynamic frames, which are different from each other, the assumptions of the attack are irrelevant. As a result, the method becomes robust against the attack.

Extraction Result
In this step, the watermarks are extracted from frames. To present the result the extracted watermark from the same sample frame, which is presented in previous steps, is chosen. The extracted watermark is shown in Figure 18.

![Figure 18: Extracted watermark from video number 4](image)

Error Checking Results
In this step, the BER is calculated for each video. During the experiments, BER is calculated for each frame separately. The results show that the BER is same for all frames in each video. For example in video number 1, the BER for each frame is 0.03. The explanation is that the attack has found the location of only 3 percent of the watermark’s bits. This location finding has no logic and it happens just by chance, since the estimation of the attack is very poles apart than the watermark. The BER is expressed in percent. It means how many bits are extracted wrongly from each 100 bits. The best result will be zero and in the worst case the BER will be 100% or it can demonstrate as one. BER of one means the attack managed to find and remove the watermark completely. The BER of video samples are presented in Table 4.

<table>
<thead>
<tr>
<th>Video</th>
<th>Average BER</th>
<th>Number of frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video 1</td>
<td>0.03</td>
<td>320</td>
</tr>
<tr>
<td>Video 2</td>
<td>0.05</td>
<td>252</td>
</tr>
<tr>
<td>Video 3</td>
<td>0.01</td>
<td>366</td>
</tr>
<tr>
<td>Video 4</td>
<td>0</td>
<td>198</td>
</tr>
<tr>
<td>Video 5</td>
<td>0.03</td>
<td>291</td>
</tr>
<tr>
<td>Video 6</td>
<td>0.04</td>
<td>294</td>
</tr>
<tr>
<td>Video 7</td>
<td>0</td>
<td>323</td>
</tr>
<tr>
<td>Video 8</td>
<td>0</td>
<td>258</td>
</tr>
<tr>
<td>Video 9</td>
<td>0.03</td>
<td>600</td>
</tr>
<tr>
<td>Video 10</td>
<td>0.01</td>
<td>453</td>
</tr>
<tr>
<td>Video 11</td>
<td>0.01</td>
<td>390</td>
</tr>
<tr>
<td>Video 12</td>
<td>0.05</td>
<td>392</td>
</tr>
<tr>
<td>Video 13</td>
<td>0.02</td>
<td>440</td>
</tr>
<tr>
<td>Video 14</td>
<td>0.01</td>
<td>367</td>
</tr>
<tr>
<td>Video 15</td>
<td>0</td>
<td>275</td>
</tr>
</tbody>
</table>

Result Analysis
This section consists of three main parts. The quality checking results and error checking outcomes are analysed in the first two parts respectively. In the third section, the comparison results of proposed method with other method is presented.

Quality Checking Results Analysis
The average PSNR from fifteen sample videos is 56.04. This shows that the proposed method maintain quality after embedding. It means that the changes are undetectable by human visual system. The average PSNR for video samples are depicted in Figure 19. This chart is prepared based on the PSNR result, which is calculated during the experiment and is demonstrated in Table 3.
According to Figure 19, the best PSNR is for video number 11 and the worst one is for video number 1. The only keen measurable difference between these two videos is their length. However, Figure 20 declare that there is no meaningful relation between the length and the quality. Therefore, it can be said that the proposed method in this work is independent from the duration of the video.

Figure 19: Average PSNR for each watermarked sample video

Figure 20: Video length (Second)

Error Checking Results

BER is other benchmark that is used to measure the robustness of the proposed method. The average of calculated error rate in this work is around 0.019. Figure 21 illustrates the average BER for each video sample. According to this chart, the most successful set-up for collusion attack is in video number 2 with 252 frames and video number 12 with 392 frames. Moreover, in video number 4 (198 frames), number 7 (323 frames), number 8 (258 frames) and number 15 (275 frames), the attack is absolutely miscarried. Nevertheless, theoretically collusion attack is more successful if the colluders have access to more watermarked frames. This can be proved by checking video number 4 (198 frames), number 7 (323 frames), number 8 (258 frames) and number 15 (275 frames), which have BER of zero and duration of less than 10 seconds. Nonetheless, video number 2 (252 frames) also has duration less than 10 seconds but the BER is 0.05. This means the theory is not absolute and it only improves the chance of estimation if there are many watermarked frames available for colluders. In other words, the robustness of this proposed method is unconnected to the duration of the video (number of frames). Figure 21 is a chart view of the BER results, which are calculated during the experiment and are demonstrated in Table 4.

Figure 21: Average BER for each sample video

Comparison of Results

Table 5 is a comparison between this work and the related works on collusion resistant watermarking. Either unfortunately, most of the other works did not mention any thing about the robustness checking method or if they mention the method, there is no numerical results. The mentioned BERs in the related works are much higher than what is achieved in this work. Consequently, it can be said that the proposed method is robust against the collusion attack. In terms of quality, the average PSNR of 56.04 is achieved that seems to be the best one in comparison with the related works. The result of comparing this work and other works are shown in Table 5 and Figure 22.

Table 5: Comparison of results

<table>
<thead>
<tr>
<th>Author</th>
<th>Year</th>
<th>PSNR (dB)</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mohamed[26]</td>
<td>2006</td>
<td>45</td>
<td>0.08</td>
</tr>
<tr>
<td>Ujwala[28]</td>
<td>2010</td>
<td>37</td>
<td>3.5</td>
</tr>
<tr>
<td>Ge [29]</td>
<td>2010</td>
<td>38</td>
<td>N/A</td>
</tr>
<tr>
<td>Havyarimana[43]</td>
<td>2011</td>
<td>38.8</td>
<td>N/A</td>
</tr>
<tr>
<td>Koubaa[31]</td>
<td>2012</td>
<td>31.5</td>
<td>N/A</td>
</tr>
<tr>
<td>Maity[32]</td>
<td>2013</td>
<td>40.37</td>
<td>N/A</td>
</tr>
<tr>
<td>Karmakar[15]</td>
<td>2015</td>
<td>43.99</td>
<td>N/A</td>
</tr>
<tr>
<td>Proposed Method</td>
<td>2016</td>
<td>56.04</td>
<td>0.019</td>
</tr>
</tbody>
</table>

N/A: Not Available

Figure 22: Quality comparison

Conclusion

This work proposed a frame-by-frame collusion resistant blind watermarking method to enhance copyright protection of digital videos. To fulfill the robustness against the collusion attack, each frame receives a different watermark. Subsequently, it is impossible for the attacker to estimate it when each frame has a unique watermark.
Moreover, to attain the robustness as well as preserving the quality, a combination of BTC, DWT and SVD methods are used. The results of the work show a significant improvement in terms of quality beside the robustness. Future work can be concentrated on the integrity that is the other aspect of copyright protection. If the embedded watermark is changed due to malicious attempts, the tampered video becomes inauthentic. Therefore, adding tamper detection can improve the copyright protection that can be considered as the future work of this effort.

Acknowledgment
The authors would like to acknowledge Universiti Teknologi Malaysia (UTM) for providing facilities and resources to get this work done. Also many thanks to Ministry of Higher Education (MOHE) for funding and supporting this research and providing excellent research environment in which this work was conducted and the scientific research targets fully accomplished (Grant no. R. K130000. 7838. 4F643).

References


