Development and Implementation of a Video Watermarking Method Based on DCT Transform

Ali Benziane[1], Suryanti Awang[2], and Mohamed Lebcir[2]
[1]Faculty of Science and Technology, University of Djelfa, Algeria
[2]Faculty of Computing, Universiti Malaysia Pahang, Malaysia

Abstract: *This paper presents a new color video watermarking technique based on the one-dimensional Discrete Cosine Transform (DCT). This approach uses a differential embedding technique to insert the bits of the watermark into the video frames so that the extraction process is blind and straightforward. To further ensure the security of the method, the binary image watermark is scrambled using Arnold transform before embedded into the video segment. Also, a color space transformation from Red, Green and Blue (RGB) to YUV is performed in order to deal with the color nature of the video segments. The proposed approach exhibits good robustness against a wide range of attacks such as video compression, cropping, Gaussian filtering, and noise adding. Finally, we propose an implementation of the video watermarking technique using the Raspberry Pi 3 platform. Nearly the same remarks may be made as in the simulation resultsconcerning the robustness against video compression attacks.*

Keywords: *Blind video watermarking, DCT, differential embedding, Raspberry Pi.*

Received May 1, 2019; accepted April 8, 2020
https://doi.org/10.34028/iajit/18/2/2

1. Introduction

The recent advances in mobile communication technologies, video sharing websites, and social networks, have made it easier to share and distribute any content [1]. To some purposes, users able to edit and re-distribute digital multimedia contents such as images, audios and videos [3]. Digital watermarking has become a promising technology for protecting digital contents from unauthorized copying and manipulation by embedding a secret information directly into the data itself [7, 14]. Basic requirements of digital watermarking are transparency, robustness, and capacity [25]. These three requirements of watermarking are conflicting and limited by each other; improvement in any one of them, affects the other two negatively [14].

The watermark can be inserted in either the spatial or the transformed domain [12]. The transform domain schemes, which are usually more robust than the spatial domain ones, have attracted much attention in the literature [1, 7]. Examples of the transformed domains are the Discrete Fourier Transform (DFT) [15] the Discrete Cosine Transform (DCT) [9], the Discrete Wavelet Transforms (DWT) [13] and the combined transform domain [6, 11].

Digital watermarking can also be blind or non-blind based on extracting process [24]. In contrast to non-blind detection, blind detection schemes do not need the original media content to recover the inserted watermarks, and hence more practical in real application [14].

The process of video watermarking exhibits more difficulties than image watermarking and is more time consuming, because video segments contains large amount of data compared to digital images [1].

Recently, transform-based video watermarking schemes have attracted a lot of attention especially those based on DCT and DWT [14]. Li *et al*. [16], the embed the watermarking information into the last DC coefficient of the last macro block in every slice of the luminance component of the Moving Pictures Expert Group v2 (MPEG2) transport stream. Their results showed good performance especially in terms of time complexity, but the overall method is strictly limited to MPEG2 coding.

Abdi *et al*. [1], have developed a watermarking scheme for H.264 video. The watermark is embedded into the video sequence by modifying the number of nonzero-quantized AC coefficients in a 4×4 block of I frames. Their scheme showed acceptable performance in terms of robustness and payload for this video compression standard.

Preda and Vizireanu [18] proposed a video watermarking method based on multi-resolution wavelet Transform. In their method, the wavelet coefficients of the second level LH, HL and HH sub-bands, are used to embed the binary image watermark by means of a quantization process.

Wang *et al*. [23] developed a robust and real-time video watermarking algorithm for MPEG-2 compressed video. A set of histogram bins deduced from the DWT low frequency sub-band, are used to embed the binary watermark. Their scheme is especially robust against geometric distortions such as cropping, rotation, scaling, and frame dropping.

Farfoura *et al*. [10] the authors proposed a semi-fragile watermarking method for the authentication of the integrity of H.264 compressed videos. The Watermark insertion is performed by flipping the signs of nonzero DCT coefficients of candidate pairs of certain DCT-transformed blocks. The embedded video frames are selected using a spatial analysis to ensure invisibility and robustness. Their technique exhibits high robustness against content-preserving attacks but it has high sensitivity against content-changing ones.

In this paper, a new blind color video watermarking technique using 1D-DCT transform is proposed and evaluated. The watermark bits are embedded into two 1D-DCT transformed sub-vectors issued from the sub-sampling the Y channel of the YUV space of the video frames. The simple difference between the corresponding sub-vectors of the watermarked frames, leads to the complete extraction of the watermark image.

Also, we examine the possibility of implementing the proposed video watermarking technique using a real hardware device which is the Raspberry Pi 3. In order to further increase the security of the implemented scheme [4], a biometric feature (fingerprint image) is used as a digital watermark [24].

The paper is organized as follows: section 2 gives a background of the used color space which is YUV. In section 3, wedescribe the general watermarking process of the video segment. The embedding and the extracting processes of the binary watermark at the frame level are detailed in section 4. Section 5 presents the simulation results and analysis. Section 6 discusses a possible implementation of the proposed method using Raspberry Pi, while section 7 concludes the paper.

2. YUV Color Model

The first color system used for video transmission was the YIQ system developed for National Television System Committee (NTSC) video TV broadcasting and the closely linked YUV standard developed for Phase Alternating Line (PAL) [21]. In both cases, it was intended to have a luma (for luminance) Y channel that would be similar to the regular black and white TV signal, together with two lower frequency color channels (I,Q) or (U,V) [21].

The conversion of the Red, Green and Blue (RGB) color system to YUV color system could be per formed using the following system of equations [22]:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

Likewise, the RGB channels could be restored back from YUV using the following system of equations [22]:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.140 \\ 1 & -0.395 & -0.581 \\ 1 & 2.032 & 0 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix} \quad (2)$$

In this paper, we have used the *Y* channel of the YUV color system to embed the watermark in order to facilitate the implementation and to ensure the robustness of the extraction procedure [6].

3. The General Embedding and Extracting Process

The proposed video watermarking method is an extended version of the image watermarking method proposed in [5]. It comprises different modules such as watermark pre-processing (scrambling), video pre-processing (frame selection and color space conversion), watermark embedding, and watermark extraction. The general embedding process is shown in Figure 1 and it includes the following steps:

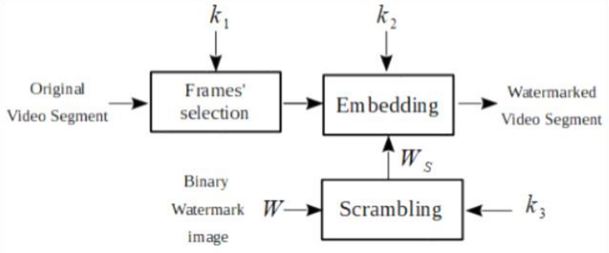


Figure 1. The general video embedding process.

- *Step* 1: Get an uncompressed video segment and convert it into frames.
- *Step* 2: Select a random set of frames to insert the watermark in, using a secret key (k_1). The number of the selected frames must be equal to the number of columns of the binary image watermark (*W*).
- *Step* 3: Scramble the binary watermark image using Arnold transform [20] and the secret key (k_3). Given a $H \times H$ square image, one of its discretized versions [13] is given as follows:

$$\begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} 1 & a \\ b & ab+1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} \bmod(H) \quad (3)$$

Where a and b are the Arnold transform parameters that can be defined by the system's user to encrypt the watermark image. So the secret key (k_3) is composed of three parameters: *a*,*b* and the number of iteration *i*.

- *Step* 4: Embed each column of the scrambled watermark image in the Y channel of the YUV space representation of the selected frame (Step 2) using the differential embedding method described in [5].

- *Step* 5: Concatenate all frames (watermarked and not) into one watermarked video segment.

On the other hand, the process of extracting the binary watermark image from the watermarked (and possibly attacked) video segment is inversely analogous to the embedding process as depicted in Figure 2.

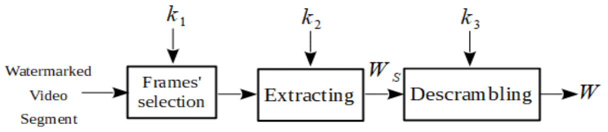


Figure 2. The general video extracting process.

- *Step* 1: Using the secret key (k_1), select the set of frames that have been used in the watermark embedding phase.
- *Step* 2: Extract each column of the scrambled watermark image from the selected frame using the extracting process described in [5]. The concatenation of all extracted columns constitutes the scrambled watermark W_s.
- *Step* 3: De-scramble the scrambled watermark image W_s using the inverse Arnold transform and the secret key (k_3) to produce the original binary watermark image W .

4. Proposed DCT-based Video Watermarking Method

The basic process of this frame embedding technique is given in Figure 3 and it includes the following sub-steps:

1. Convert the input frame from RGB to YUV (the luminance Y and two chrominance components U and V) color space and get the intensity Y channel matrix of size $M \times N$ using Equation (1).
2. Perform zigzag scanning on the Y channel matrix to convert it into one vector z.
3. Generate two sub-vectors z_1 and z_2 from the vector y using the following sub-sampling operation:

$$z_1(k) = z(2k) \tag{4}$$

$$z_2(k) = z(2k-1) \tag{5}$$

Where $k=1,\ldots,M \times N$.

4. Perform DCT on z_1 and z_2 to produce their DCT-transformed versions Z_1 and Z_2.
5. Insert the vector of the watermark bits (which is one column of W_s) in random locations (chosen using a secret key k_2) of the newly produced sub-vectors Z_1 and Z_2 as follows:

$$\hat{Z}_1 = \frac{1}{2}[Z_1 + Z_2] + \alpha W_s \tag{6}$$

$$\hat{Z}_2 = \frac{1}{2}[Z_1 + Z_2] - \alpha W_s \tag{7}$$

6. Perform the inverse DCT on $\hat{Z}_1$ and $\hat{Z}_2$ to obtain a modified sub-vectors $\hat{z}_1$ and $\hat{z}_2$.
7. Produce one modified vector $\hat{z}$ by merging the new obtained sub-vectors $\hat{z}_1$ and $\hat{z}_2$ using the inverse sub-sampling operation given in Equations (3) and (4).
8. Perform the inverse of the previously used zigzag scan operation (Step 2) to re-convert the watermarked vector $\hat{z}$ into the new matrix of intensity channel $\hat{Y}$.
9. Construct the watermarked *RGB* frame from the watermarked intensity channel $\hat{Y}$ and the two original color channels (U, V) using Equation (2).

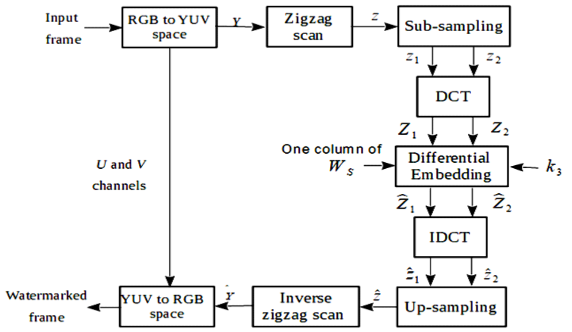


Figure 3. The embedding process for video frames.

Notice that if the input frame is a watermarked one, and by analogy with the embedding process, the extraction process will produce one watermark vector as shown in Figure 4.

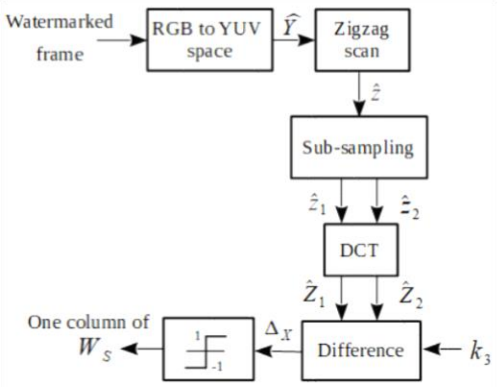
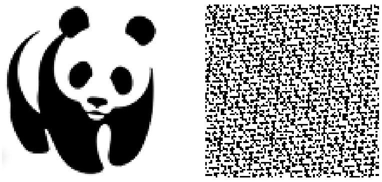


Figure 4. The extraction process of the DCT method.

5. Experimental Results

The proposed algorithm is evaluated for five Audio Video Interleaved (AVI) format uncompressed video sequences[1] of size 300 frames of 352x288: ‘City’, ‘Coastguard’, ‘Crew’, ‘Foreman’, and ‘Soccer’. The chosen watermark is a binary image of size 100×100 shown in Figure 5. Also, we set $\alpha = 0.3$ as the default gain factor value as suggested in [5].

[1] https://media.xiph.org/video/derf/



a) The original. b) The Scrambled.

Figure 5. Scrambling of the watermark image using Arnold transform.

Toassess the quality of watermarked video segments, the average PSNR measure is used [19]:

$$PSNR = 10 \times \log_{10}\left(\frac{3 \times 255^2}{MSE_R + MSE_G + MSE_B}\right) \quad (8)$$

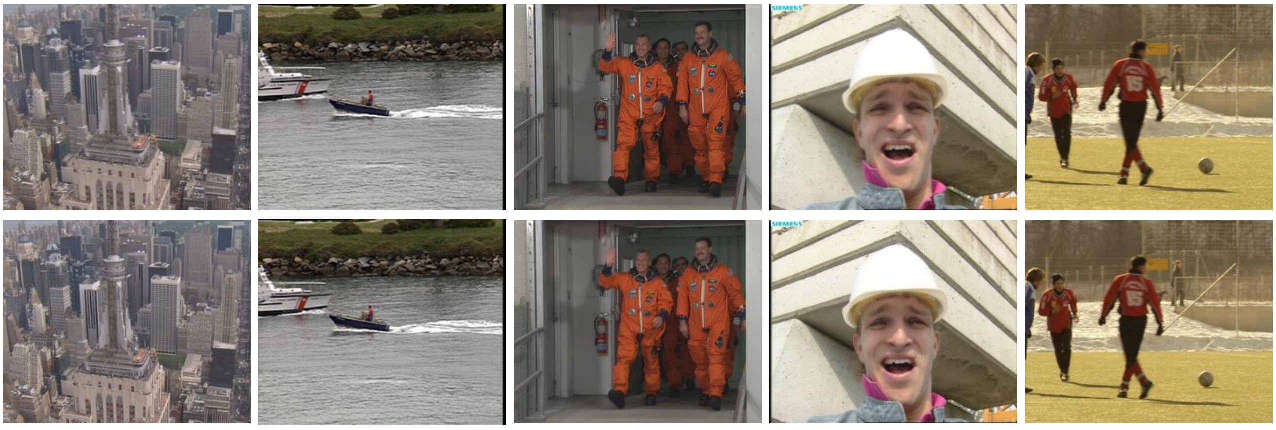
$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left(u(i,j) - \hat{u}(i,j)\right)^2 \quad (9)$$

Where $u(i,j)$ and $\hat{u}(i,j)$ are respectively, the original and the watermarked intensities belonging to R, G and B planes of the video frame.Moreover, the Bit Correct Rate (BCR) is chosen to assess the similarity between extracted and the original watermarks [5]:

$$BCR = \frac{1}{M \times N}\sum_{j=0}^{M-1}\sum_{k=0}^{N-1}\overline{W(j,k) \oplus \tilde{W}(j,k)} \times 100\% \quad (10)$$

Where $\tilde{W}$ and W are respectively the extracted and the original watermarks of sizes $M \times N$ and $\oplus$ is the binary XOR operator.

Figure 6 shows original video frames as well as the watermarked ones for the five test video segments. It is obvious that the quality of the watermarked frames is preserved qualitatively and quantitatively.



a) City. b) Coastguard. c) Crew. d) Foreman. e) Soccer.

Figure 6. Original (Top) and watermarked (Bottom) frames. The average PSNR value is about 41dB for all these video segments.

5.1. Robustness Test

5.1.1. Robustness Against Common Attacks

The robustness of the proposed video watermarking method against noise addition, Gaussian filtering, Gamma correction, sharpening and cropping is evaluated in Tables 1 and 2. Table 3 shows the extracted watermarks after different types of high-strength attacks for the video segment of 'soccer'. It's obvious that the suggested DCT-based video watermarking method is robust against all the above attacks even for high intensity ones like the case of Gaussian noise of high variance.

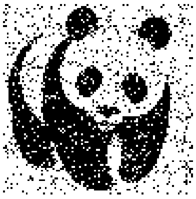
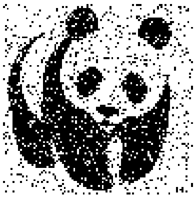
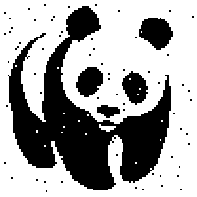
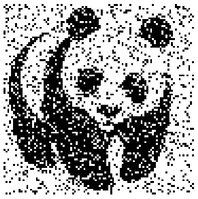
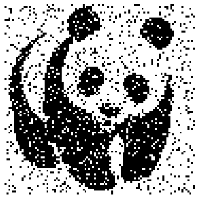
Table 1. Robustness of the proposed video watermarking technique against noise adding attacks.

Attack	Video segment				
	city	coastguard	crew	foreman	soccer
Gaussian noise (var=0.01)	100	100	100	100	100
Gaussian noise (var=0.02)	99.8	99.8	99.8	99.7	99.8
Gaussian noise (var=0.03)	99.3	99.0	99.3	99.0	99.1
Salt & pepper noise (var=0.01)	100	100	100	100	100
Salt & pepper noise (var=0.02)	100	100	100	100	100
Salt & pepper noise (var=0.03)	100	100	100	100	100

Table 2. Robustness of the proposed video watermarking technique against different types of attacks.

Attack	Video segment				
	city	coastguard	crew	foreman	soccer
Gaussian filter (5x5) var=1	100	100	100	100	100
Gaussian filter (3x3) var=1	100	100	100	100	100
Gamma correction (2)	100	100	100	100	100
Laplacian sharpening	100	100	100	100	100
Surrounding crop (10%)	100	100	100	100	100
Surrounding crop (25 %)	100	100	100	100	100

Table 3. The extracted watermark under different types of attacks.

Gaussian noise (var =0.1)	Motion JPEG (Q=40)	Cropping (30%)	MPEG4-H.264 (Q=70)	Motion JPEG2000 (R=50)
BCR=90.25 %	BCR=90.01 %	BCR=99.1 %	BCR=84.1 %	BCR=89.0 %
				

5.1.2. Robustness Against Compression Attacks

To assess the robustness against video compression, the ‘Motion JPEG’, ‘Motion JPEG2000’, and MPEG-4 ‘H.264’ compression standards are used. The plot of the results of robustness against these attacks is shown in Figures 7, 8, and 9 respectively.

One can see clearly from these results that the presented watermarking algorithm exhibits very good performance against motion JPEG and JPEG2000 compression standards. Differently, it shows a lower performance against MPEG-4 attacks due to the very high compression rate of this standard.

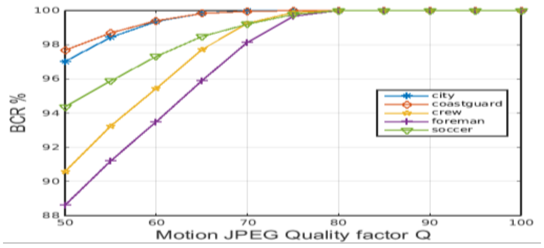


Figure 7. Robustness against Motion JPEG attack.

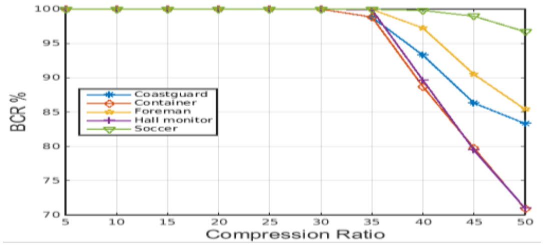


Figure 8. Robustness against Motion JPEG2000 attack.

6. Implementation Using Raspberry Pi

In this section, we deal with the implementation of the proposed video watermarking technique using Raspberry Pi (RP) platform [2].

Instead of using an arbitrary binary watermark one can effectively use a fingerprint image (of size 200×200) issued from a fingerprint scanner to watermark the video segments. Therefore, the watermarking process will be much more secure because the fingerprint patterns are highly person-dependent [2].

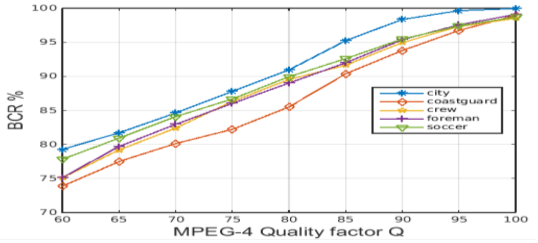


Figure 9. Robustness against MPEG-4 H-264 compression attack.

6.1. The Implementation Strategy

The propose dimplementation of the video watermarking system is shown in Figure 10. The Raspberry Pi 3 platform running Python and OpenCV on Raspbian OS, makes the implementation straight forward [17].

Along with Raspberry Pi board, we used a camera module to capture the video segments, a fingerprint scanner to retrieve the fingerprint image and a Liquid Crystal Display (LCD) screen to provide a mean of interaction with the developed Graphical User Interface (GUI) as shown in Figure 12.

The implementation of the general video DCT-based video watermarking using the RP platforme requires two major adaptation which are:

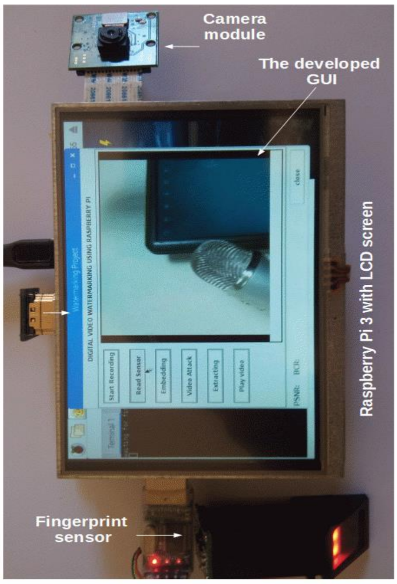


Figure 10. The proposed implementation of the video watermarking system.

- The zig-zag scanning in Figures 3 and 4 is replaced by a simple column-wise concatenation of each frame in order to transform the channel Y (resulted from RGB to YUV conversion) into one vector. This modification is needed in order to enhance the speed of the video watermarking process on RP. This will affect slightly the transparency of the watermarked video segment, because of the low correlation between the sub-vectors Z_1 and Z_2 and (section 3) in this case, but this drawback can be compensated by embedding fewer number of bits in each frame.

- The watermark which is a fingerprint image has to be binarized (Figure 11) before embedding into the video segment. A hard thresholding binarization with a threshold of 127 is used in the experiments. After that, the general embedding process as in Figure 1 is performed by the RP platform using the new binary watermark.

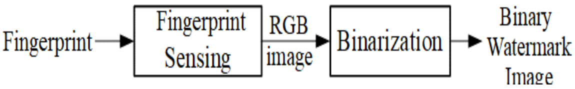
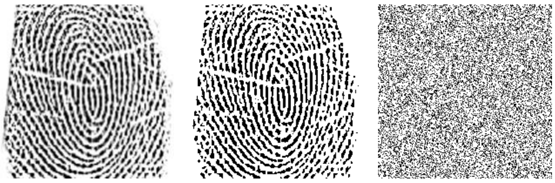


Figure 11. Preprocessing of the fingerprint image.



a) Original. b) Binarized. c) Scrambled.

Figure 12. Example results of the preprocessing of the fingerprint image.

6.2. Implementation Performance

In the experiments on the implemented system, video segments of 300 frames are captured (by the camera module) with 288×252 resolution at 30 frames per second in order to be compatible with those used in the general DCT-based algorithm.

To verify the robustness of the implemented algorithm, we concentrate only on following video compression attacks: Motion JPEG (MJPEG), MPEG-4, and H.264 standards because of their availability on the RP operating system. Notice that the PSNR of the watermarked video are preserved through the embedding factor α to be slightly more than 35dB.

6.2.1. Execution Speed

The processing time per function for the watermark embedding and extracting (given by the Python Profilers module) are shown by Tables 4 and 5 respectively.

Table 4. Execution time for watermark embedding.

Function	frm_embed	vid_embed	dct	idct
Total time (s)	6.05	27.32	6.03	5.75
N °of calls	200	1	400	400
Time per call (s)	0.03	27.32	0.01	0.01

Table 5. Time execution for watermark extracting.

Function	frm_extract	dct	vid_extract
Total time (s)	6.34	4.02	11.73
N °of calls	200	400	1
Time per call (s)	0.03	0.01	11.73

The watermark embedding at the frame level (embedding one column of the watermark image) takes only 0.03s which is acceptable. The same remark can be made regarding the watermark extracting at the frame level. Also, the most time-consuming functions at the watermark embedding and extracting phases are the DCT and the inverse DCT which is consistent with the features of transform domain watermarking techniques.

Note that the total watermark embedding and extracting times (27.32s and 11.73s respectively) can be reduced significantly by using FPGA-based system instead of Raspberry Pi but at the expense of the implementation complexity [12].

6.2.2. Robustness Against Compression

Tables 6 and 7 shows respectively the BCR of the extracted watermark for MJPEG and MPEG4 compression standards of three video segments and for different quality scale. Table 8 shows the performance of the implemented method against the more recent and size-efficient type of video compression, i.e, the H.264 standard. In this type of compression the Constant Rate Factor (CRF) is used as quality assessment measure [8].

Table 6. Robustness against Motion JPEG compression attack.

Quality (Qscal)	1	2	3	4	5
Segment 1	100	100	100	100	99
Segment 2	100	100	100	100	99
Segment 3	100	100	100	100	99

Table 7. Robustness against MPEG4 compression attack.

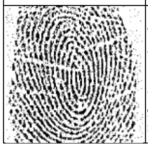
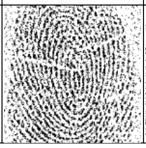
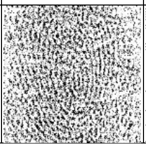
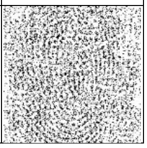
Quality (Qscal)	1	2	3	4	5
Segment 1	100	100	100	99	97
Segment 2	100	100	100	99	98
Segment 3	100	100	100	98	96

Table 8. Robustness against H.264 compression attacks.

CRF	2	5	10	15	20
Segment 1	100	100	100	99	87
Segment 2	100	100	100	99	84
Segment 3	100	100	100	98	83

These results confirm the observations made in section 4 about the robustness of the proposed method against video compression even with the usage of column wise concatenation instead of the zigzag operation. The robustness against Motion JPEG and MPEG-4 standards is high for the acceptable range of the quality factor (Qscal) used in the implementation. For the H264 standard, the behavior of the implemented method is also similar to the simulation results given in section 4; it has very good performance under low compression ratios but it gets limited under high to severe ones as shown in Table 9.

Table 9. The extracted watermark under H.264 compression attacks.

CRF=20	CRF=22	CRF=24	CRF=26
BCR=96 %	BCR=86 %	BCR=77 %	BCR=75 %
			

7. Conclusions

In this paper we have designed and implemented a blind and robust watermarking technique for uncompressed video using the DCT transform. The technique is based on the differential embedding of the bits of the watermark in DCT domain. The perceptual quality of the watermarked video segments and the average PSNR values have been very good for all tested video segments.Different types of attacks have been performedon the watermarked video segments and the corresponding BCR values have shown that the proposed technique fairly fulfills the robustness requirements.

Furthermore, the Raspberry Pi 3 platform is employed to implement a reduced version of the proposed video watermarking method.In this implementation, the user fingerprint image is used as the binary watermark and the zig-zag operation is replaced with column-wise concatenation in order toaccelerate the embedding and extracting processes. The experimental results have validated those found in the simulation and have proved the feasibility of the implementation on the Raspberry Pi platform.

Acknowledgments

We would like to show our gratitude to Universiti Malaysia Pahang (RDU vote number RDU190315) for supporting this study.

References

[1] Abdi L., Ben Abdallah F., and Meddeb A., "Real-Time Watermarking Algorithm of H.264/AVC Video Stream," *The International Arab Journal of Information Technology*, vol. 14, no. 2, pp. 168-174, 2017.

[2] Abughalieh K., Sababha B., and Rawashdeh N., "A Video-Based Object Detection and Tracking System for Weight Sensitive UAVs," *Multimedia Tools and Applications*, vol. 78, no. 7, pp. 9149-9167, 2019.

[3] Al-Saffar A., Awang S., Tao H., Omar N., Al-Saiagh W., and Al-bared M., "Malay Sentiment Analysis Based on Combined Classification Approaches and Senti-Lexicon Algorithm," *PloS one*, vol. 13, no. 4, pp. e0194852, 2018.

[4] Awang S., Yusof R., Zamzuri M., and Arfa R., "Feature Level Fusion of Face and Signature Using A Modified Feature Selection Technique," *in Proceedings of the International Conference on Signal-Image Technology and Internet-Based Systems*, Kyoto, pp. 706-713, 2013.

[5] Benoraira A., Benmahammed K., and Boucenna N., "Blind Image Watermarking Technique Based on Differential Embedding in DWT and DCT Domains," *EURASIP Journal on Advances in Signal Processing*, vol. 2015, no. 1, pp. 1-11, 2015.

[6] Bhagyashri S. and Joshi M., "All Frequency Band DWT-SVD Robust Watermarking Technique for Color Images in YUV Color Space," *in Proceedings of the International Conference on Computer Science and Automation Engineering*, Shanghai, pp. 295-299, 2011.

[7] Bhardwaj A., Verma V., and Jha R., "Robust Video Watermarking Using Significant Frame Selection Based on Coefficient Difference of Lifting Wavelet Transform," *Multimedia Tools and Applications*, vol. 77, no. 15, pp. 19659-19678, 2018.

[8] Dey B. and Kundu M., "Robust Background Subtraction for Network Surveillance in H.264 Streaming Video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 10, pp. 1695-1703, 2013.

[9] Dutta M., Singh A., and Soni K., "A Secure Algorithm for Biometric-Based Digital Image Watermarking in DCT Domain," *International Journal of Computational Vision and Robotics*, vol. 4, no. 12, pp. 99-114, 2014.

[10] Farfoura M., Horng S., Guo J., and Al-Haj A., "Low Complexity Semi-Fragile Watermarking Scheme for h.264/avc Authentication," *Multimedia Tools and Applications*, vol. 75, no. 13, pp. 7465-7493, 2016.

[11] Hu H. and Hsu L., "Collective Blind Image Watermarking in DWT-DCT Domain with Adaptive Embedding Strength Governed by Quality Metrics," *Multimedia Tools and Applications*, vol. 76, no. 5, pp. 6575-6594, 2016.

[12] Joshi A., Mishra V., and Patrikar R., "FPGA Prototyping of Video Watermarking for Ownership Verification Based on H.264/AVC," *Multimedia Tools and Applications*, vol. 75, no. 6, pp. 3121-3144, 2016.

[13] Khalili M. and Asatryan D., "Colour spaces Effects on Improved Discrete Wavelet Transform-Based Digital Image Watermarking Using Arnold Transform Map," *IET Signal Processing*, vol. 7, no. 3, pp. 177-187, 2013.

[14] Khan A., Siddiqa A., Munib S., and Malik S., "A Recent Survey of Reversible Watermarking Techniques," *Information Sciences*, vol. 279, pp. 251-272, 2014.

[15] Kumari V. and Thanushkodi K., "A Secure Fast 2d-Discrete Fractional Fourier Transform Based Medical Image Compression Using SPIHT Algorithm with Huffman Encoder," *International Review on Computers and Software*, vol. 8, no. 7, pp. 1702-1710, 2013.

[16] Li J., Wang Y., and Dong S., "Video Watermarking Algorithm based DC Coefficient," *in Proceedings of the International Conference on Image Vision and Computing*, Chengdu, pp. 454-458, 2017.

[17] Marot J. and Bourennane S., "Raspberry Pi for Image Processing Education," *in Proceedings of the European Signal Processing Conference*, Kos, pp. 2364-2366, 2017.

[18] Preda R. and Vizireanu D., "Robust Wavelet-Based Video Watermarking Scheme for Copyright Protection Using The Human Visual System," *Journal of Electronic Imaging*, vol. 20, no. 1, pp. 13022-13022, 2011.

[19] Rupp M., *Video and Multimedia Transmissions over Cellular Networks*, Wiley, 2009.

[20] Saadi S., Merrad A., and Benziane A., "Novel Secured Scheme for Blind Audio/Speech Norm-Space Watermarking by Arnold Algorithm," *Journal of Signal Processing*, vol. 154, pp. 74-86, 2019.

[21] Szeliski R., *Computer Vision: Algorithms and Applications*, Springer, 2011.

[22] Tkalcic M. and Tasic J., "Colour Spaces: Perceptual, Historical and Applicational Background," *in Proceeding of EUROCON Computer as a Tool*, Ljubljana, pp. 304-308, 2003.

[23] Wang L., Ling H., and Lu Z., "Real-Time Compressed-Domain Video Watermarking Resistance to Geometric Distortions," *IEEE MultiMedia*, vol. 19, no. 1, pp. 70-79, 2012.

[24] Wójtowicz W. Ogiela M., "Digital Images Authentication Scheme Based on Bimodal Biometric Watermarking in an Independent Domain," *Journal of Visual Communication and Image Representation*, vol. 38, pp. 1-10, 2016.

[25] Yu X., Wang C., and Zhou X., "A Survey on Robust Video Watermarking Algorithms for Copyright Protection," *Applied Sciences*, vol. 8, no. 10, pp. 1-26, 2018.

Ali Benziane received a Ph. D degree in Electronics in 2017 from the University of Setif, Algeria. He is presently working *as* an Assistant Professor *at* the Faculty of Science and Technology, University of Djelfa, Algeria. His field of interest includes Image Processing, Embedded Systems and IoT.

Suryanti Awang received Ph. D degree in Electrical Engineering in 2014 from UniversitiTeknologi Malaysia, Johor Bahru, Malaysia. Dr. Suryanti is currently a senior lecturer at Universiti Malaysia Pahang, Malaysia from 2005 until now. Her research interests include Pattern Recognition, Machine Learning, and Soft Computing.

Mohamed Lebcir received a master's degree in Computer Science (Option: Image and Speech processing) in 2015 from University of Djalfa, Algeria. He is currently a PhD Student at Universiti Malaysia Pahang, Malaysia. His research interests include Pattern Recognition and Image and Audio Processing.