EXCELLENT PUBLISHERS

International Journal of Current Research and Academic Review

ISSN: 2347-3215 Volume 4 Number 3 (March-2016) pp. 1-15

Journal home page: http://www.ijcrar.com

doi: http://dx.doi.org/10.20546/ijcrar.2016.403.001

Video Watermarking Using Wavelet Transform

Hanan Qassim Jaleel[1]* and Jane Jaleel Stephan[2]

[1]Baghdad College of Pharmacy, Iraq
[2]University of Information Technology and Communication (UOITC), Iraq
Corresponding author

KEYWORDS

watermarking, information hiding, security, video watermarking and wavelet transform

ABSTRACT

In this paper, a robust video watermarking scheme is proposed to provide copyright protection of the video using wavelet transform as embedding domain. The uncompressed AVI video frames are converted to bitmap images and transformed to the Haar wavelet domain in order to embed the watermark. The embedding technique used is least significant bit (LSB) which is improved by using a pseudo random sequence generated by Linear Feedback Shift Register (LFSR). This sequence is used to obtain pixel positions in order to spread and embed watermark slice bits in each video frame. The proposed system implements five types of attack (Frame Dropping, Frame Swapping,Salt and Pepper Noise, Frame Rotating and Frame Cropping) on the tested video samples and the experimental results have shown that the embedded watermark is resistant to them with good measurement values.

Introduction

Digital video is readily reproduced and distributed over information networks. However, these attractive properties lead to problems enforcing copyright protection. As a result, creators and distributors of digital video are hesitant to provide access to their digital intellectual property. Digital watermarking has been proposed as a means to identify the owner and distribution path of digital data. Digital watermarks address this issue by embedding owner identification directly into the digital data itself. The information is embedded by making small modifications to the pixels in each video frame. When the ownership of a video is in question, the information can be extracted to completely characterize the owner or distributor of the data (Doeerr and Dugelay, 2003; Pik-Wah, 2004).

A watermark is a digital data embedded in multimedia objects such that the watermark can be detected or extracted at later times in order to make an assertion about the object. The main purpose of digital watermarking is to embed information imperceptibly and robustly in the host data. Typically the watermark contains information about the origin, ownership, destination, copy control, transaction etc. Potential applications of digital watermarking include transaction tracking, copy control, authentication, legacy system enhancement and database linking etc (Pu *et al*., 2010; Essouabi *et al*., 2009; Abdulfetah *et al*., 2010).

Video Watermarking Techniques

Apparently any image watermarking technique can be extended to watermark videos, but in reality video watermarking techniques need to meet other challenges than that in image watermarking schemes such as large volume of inherently redundant data between frames, the unbalance between the motion and motionless regions, real-time requirements in the video broadcasting etc. Watermarked video sequences are very much susceptible to pirate attacks such as frame averaging, frame swapping, statistical analysis, digital-analog (AD/DA) conversion, and lossy compressions. Video watermarking applications can be grouped as security related like Copy control, fingerprinting, ownership identification, authentication, taper resistance etc. or value added applications like legacy system enhancement, database linking, video tagging, digital video broadcast monitoring, Media Bridge etc.

Apart from robustness, reliability, imperceptibility, practicality, video watermarking algorithms should also address issues such as localized detection, real time algorithm complexity, synchronization recovery, effects of floating point representation, power dissipation etc.

According to the working domain, video watermarking techniques are classified in pixel domain and transform domain techniques. In pixel domain the watermark is embedded in the source video by simple addition or bit replacement of selected pixel positions. The main advantages of using pixel domain techniques are that they are conceptually simple to understand and the time complexity of these techniques are low which favours real time implementations. But these techniques generally lacks in providing adequate robustness and imperceptibility requirements. In transform domain methods, the host signal is transformed into a different domain and watermark is embedded in selective coefficients. Commonly used transform methodologies are discrete cosine transformation (DCT) and discrete wavelet transformation (DWT). Detection is generally performed by transforming the received signal into appropriate domain and searching for the watermarking patterns or attributes. The main advantage of the transformed domain watermarking is the easy applicability of special transformed domain properties. For example, working in the frequency domain enables us to apply more advanced properties of the human visual system (HVS) to ensure better robustness and imperceptibility criteria (Doeerr and Dugelay, 2003; Bhattacharya *et al*., 2006; Chetan and Raghavendra, 2010).

Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT), which is based on sub-band coding is found to yield a fast computation of Wavelet Transform. It is easy to implement and reduces the computation time and resources required.

DWT can be used for image watermarking embedding. The DWT (Discrete Wavelet Transform) separates an image into a lower resolution approximation image (LL) as well as horizontal (HL), vertical (LH) and diagonal (HH) detail components (Bhattacharya *et al*., 2006; Chetan and Raghavendra, 2010). The process can then repeated to compute multiple “scale” wavelet decomposition (Hassan, 2006), as in fig (3). Each sub band is then decomposed again. Finally, after performing the second level decomposition there are 16 sub-subbands of each block (Chetan and Raghavendra, 2010). However as in Fig (1) only the LH2 an HL2 sub-sub band are selected from the original decomposition. Most embedding algorithms avoid selecting the LL and HH sub-sub band as they contain low frequency data which are likely to be seen if a watermark has been inserted (Hassan, 2006). On the other-hand, the sub-subband HH contains high frequency data which are likely to be eliminated when the watermarked image is compressed (Hassan, 2006; Jayamalar and Radha, 2010).

Linear Feedback Shift Register

A linear feedback shift register (LFSR) is a shift register whose input bit is a linear function of its previous state. The only linear function of single bits is XOR, thus it is a shift register whose input bit is driven by the Exclusive OR (XOR) of some bits of the overall shift register value (George *et al*., 1999).

One of the two main parts of an LFSR is the shift register. A shift register is an easy to build and comparatively inexpensive. The shift register is a sequence of bits. The length of a shift register is figured in bits, if it is n bits long, it's called an n-bit shift register. Each time a bit is needed, all of the bits in the shift register are shifted one bit to the right, and the new left-most bit is computed as a function of the other bits in the register.

The output of the shift register is one bit, often the least significant bit. The period of a shift register is the length of the output sequence before it starts repeating. As an example, if LFSR is initialized with value (1111), it produces the following output sequence: 111101011001000 (Al-Kenany, 1999; Shi, 2010).

In an LFSR, the bits contained in selected positions in the shift register are combined in some sort of function and the result is fed back into the register's input bit. By definition, the selected bit values are collected before the register is clocked and the result of the feedback function is inserted into the shift register during the shift, filling the position that is emptied as a result of the shift. The feedback function in an LFSR has several names: XOR, odd parity, sum modulo 2. The bit positions selected for use in the feedback function are called "taps". The list of the taps is known as the "tap sequence" (Shi, 2010; Von, 2004).

The Proposed Video Watermarking Scheme

The proposed video watermarking system embeds a watermark image in the AVI cover video file for the copyright protection of video file. The proposed system consists generally of two modules, the first module for embedding, and the second module for extracting.

The proposed scheme is able to deal with AVI uncompressed video files and gray bitmap images with sizev(64 * 64)pixels as watermark images**.** The watermark image bits are stored in a buffer and divided into slices such that each slice bits are embedded in the LSB of the wavelet coefficients in pixel positions generated by LFSR sequence in each decomposed image. The watermark slice size is calculated according to the formula

$$\textbf{Watermark Slice Size} = \frac{\textbf{Watermark Size (in bits)}}{\textbf{No. of Frames in Video}}$$

The watermark slice size is calculated in the embedding process to determine the slice size of bits that should be embedded in each decomposed image. The proposed system embeds the watermark image size in the first decomposed image in order to be extracted later by extraction process, gets the watermark size and calculates the watermark slice size that should be recovered from each decomposed image. The scheme extracts video frames from video stream, converts video frames to images, then transform them to the wavelet domain in order to embed watermark slice bits.

The proposed watermarking scheme decomposes the bitmap image into regions using the haar wavelet decomposition technique. It decomposes the image into four subbands LL, LH, HL and HH. In the proposed scheme, image decomposition does not go beyond the one level.

The watermark embedding in each frame will be in the embedding window with size (M * N) pixels in the (LH) subband, where M and N represents the dimensions of the subband. The embedding has been performed in the subband (LH) only which shows robustness to attacks and security of the hidden watermark. The scheme embeds watermark slices in all decomposed images corresponding to video frames in an embedding window in the (LH) subband of each image. At the beginning, the system calculates the size of the (LH) subband of each decomposed image and gets the subband dimensions denoted by M and N, then the system calculates the embedding window size in order to separate and embed watermark bits in it.

The proposed system generates a random binary number whose length is equal to M+N, and this binary stream of bits becomes an initial state(seed) to the Linear Feedback Shift Register LFSR which generates pseudo random sequence that represents positions to embed the watermark slice in the wavelet middle frequency subband (LH) of each decomposed image. The watermark slice is embedded in an embedding window in the subband of each image and PSNR and other distortion measures MSE and MAE values are evaluated. The proposed scheme embeds the initial (seed) binary string itself in each decomposed image in order to be extracted later by extraction process and generates the same LFSR sequence and obtains the same locations to recover the watermark slice bits from each decomposed image. Also, the scheme embeds watermark size in order to be extracted in extraction and used to calculate watermark slice size.

The proposed scheme generates a random binary string. The length of the string is equal to the length of linear feedback shift register LFSR since each binary string will be an initial seed to the shift register to generate the sequence in order to obtain positions used for embedding in each decomposed image. The length of the linear feedback shift register is calculated according to the dimensions of the embedding window which is used to spread and embed the watermark slice bits.

For example, Suppose that the dimension of each image corresponding to video frame is (320 × 240) pixels, then the dimensions of the (LH) subband will be (160 × 120) pixels. The embedding window dimensions in the LH subband is (128 × 64) pixels, where the window width is equal to 128 and denoted by symbol M and the window height is 64 and denoted by symbol N. The linear feedback shift register is denoted by symbol R and the length calculation equation will be

R= Log M + Log N ………………... (2)

R= log 128 + log 64

R$\cong$ log 2^7 + log 2^6
R= 7 + 6 =13 bits

The LFSR length will be 13 bits depicted in Figure (2). The source tap is determined as the first location (bit) and the destination tap is determined as the last location (bit) in the LFSR.

So the string length will be 13 binary bits generated randomly. Assume that the initial state (seed) for LFSR is 1110101100100, then the LFSR generator corresponding to the binary string and some of the generated internal states as pixel positions for embedding are depicted in Table (1).

Any internal state of LFSR is used to create a pixel position in the embedding window. The least six bits of the LFSR state represent the column number, where the most seven bits represent the row number. For this example, the maximum row number that can be obtained is 128 and the maximum column number is 64 in each state. These pixels represent positions to spread and embed watermark slice bits in the embedding windows.

The Embedding Stage

The embedding of watermark slice in each decomposed image is performed according to positions generated by LFSR to embed the watermark slice. The block diagram of the embedding stage is illustrated in figure (3) while Algorithm (1) describes the proposed Embedding Module.

After the embedding process completes, the system convert watermarked images to video frames, creates new AVI file and adds the frames in order to create the watermarked video file.

Reconstruction of Decomposed Images

The proposed embedding process applies inverse Haar wavelet transform to each decomposed image in order to return the reconstructed image and store it in the reconstructed image buffer.

The Extracting Stage

In this stage the watermark slice bits are extracted from bitmap images corresponding to video frames, stored in a buffer, accumulated to create a vector of all extracted watermark slices in order to construct the recovered watermark image. Figure (4) shows a block diagram of the Extracting stage, and the extraction module can be shown in algorithm (2).

Experimental Results

To verify the performance and effectiveness of the proposed watermarking scheme,The embedding and extracting tests were applied on AVI video samples with different watermark images. We present the results of embedding and extracting test on the AVI video sample (greenland.avi). It consists of 73 frames, the dimension of each frame are (320 × 240) pixels. The watermark image (lena.bmp) is a gray image with dimensions (64 × 64) pixels. Figure (5) shows some of the video frames and watermark image.Table (2) shows the values of PSNR, MSE and MAE after embedding watermark slices in video frames. Figure (6) shows the original and extracted watermark images. The recovered watermark is the same as the original and the comparison yields higher PSNR value for the recovered watermark image.

Watermark Attack Results

To evaluate the robustness and quality of the recovered watermark image, five types of attacks are imposed on the tested watermarked video samples illustrated as follows,

Frame Dropping

Table (3) shows the results of distortion measure values of the recovered watermark after applying frame dropping attack on the watermarked video (dubai_airport.avi) with different numbers of randomly dropped frames. The video consists of 100 frames each has the same format (BMP file) and the dimensions of each frame are (320 × 240) pixels. The watermark image (Monaliza.bmp) is a gray image with size (64*64) pixels.

Table.1 Some of Generated Internal States of LFSR as Pixel Positions for Embedding

LFSR internal states													Output Stream	Pixels Positions
1	0	1	1	1	0	0	1	0	1	0	1	1		
0	1	0	1	1	1	0	0	1	0	1	0	1	1	(46,21)
1	0	1	46		1	1	0	0	21		1	0	1	(87,10)
1	1	0	87		1	1	1	0	(	10	0	1	1	(107,37)
0	1	1	107		0	1	1	1	(	37	1	0	0	(53,50)
0	0	1	53		1	0	1	1	1	50	0	1	0	(26,57)
			26							57				

Table.2 PSNR, MSE and MAE Values after Embedding Watermark Slices in all Frames

Frame Number	MAE	MSE	PSNR
1	0.0371	0.0861	58.7831
2	0.0362	0.0829	58.9437
3	0.0377	0.0865	58.7588
4	0.0378	0.0868	58.7471
. .			
30	0.0367	0.0839	58.8935
31	0.0396	0.0934	58.4280
. .			
70	0.0360	0.0818	59.0007
71	0.0380	0.0866	58.7575
72	0.0361	0.0817	59.0069
73	0.0384	0.0868	58.7445

Table.3 Results of Distortion Measure Values of the Recovered Watermark After Applying Frame Dropping Attack

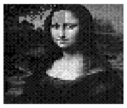
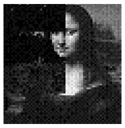
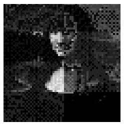
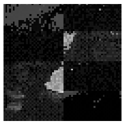
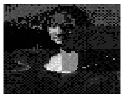
Dropped frames	Recovered watermark image	PSNR	MSE	MAE
10		28.0525	101.8196	25.7981
30		30.5822	56.8669	18.6704
50		26.6892	139.3667	30.1345
70		26.4124	148.5378	38.7644
80		25.9787	164.1387	28.7148

Figure.1 A Multiresolution Decomposition of the Image

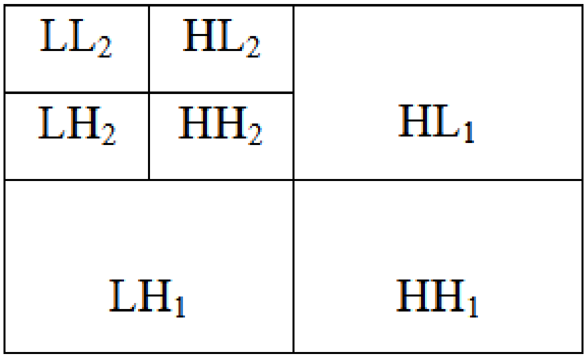


Table.4 Results of Distortion Measure Values of the Recovered Watermark after Applying Frame Swapping Attack

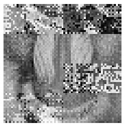
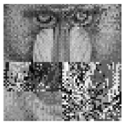
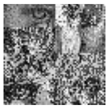
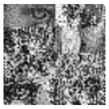
swapped frames	Recovered watermark image	PSNR	MSE	MAE
5		34.0227	25.7517	10.0935
10		31.3045	48.1533	14.8457
20		29.7106	69.5063	12.9829
30		29.0054	81.7598	22.0649
40		27.9677	103.8271	28.6040
50		28.0065	102.9026	28.7002

Figure.2 LFSR Generator

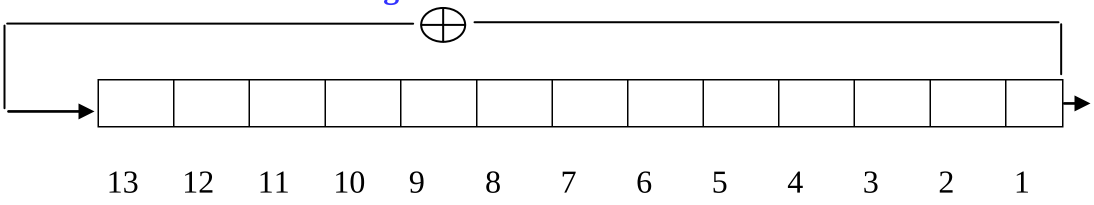


Table.5 Results of Distortion Measure Values of the Recovered Watermark after Applying Salt and Pepper Noise Attack

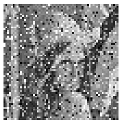
Noise Density	Extracted watermark image	PSNR	MSE	MAE
0.01		41.3387	4.7776	1.1772
0.03		37.0375	12.8628	3.1306
0.05		34.7743	21.6597	5.1860
0.07		33.7630	27.3389	6.5510
0.09		33.2660	30.6538	7.3606
0.12		32.5431	36.2051	9.0793
0.14		32.0156	40.8813	10.4753
0.16		31.4463	46.6074	12.1287

Table.6 Results of Distortion Measure Values of the Recovered Watermark after applying Frame Rotating Attack

Rotated frames	Extracted watermark image	PSNR	MSE	MAE
10		38.8495	8.4749	2.5437
30		33.6777	27.8811	6.9512
50		33.6905	27.7988	6.0659
70		32.6662	35.1931	5.0896
80		32.4088	37.3425	7.3264

Table.7 Results of Distortion Measure Values of the Recovered Watermark after Applying Frame Cropping Attack

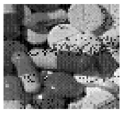
Cropping area size	Recovered watermark image	PSNR	MSE	MAE
25 × 25		41.3465	4.7690	0.6689
50 × 50		35.6037	17.8943	5.0818
75 × 75		34.0233	25.7485	7.2583
100 × 100		31.7643	43.3159	10.5532
125 × 125		30.3760	59.6316	14.0667
150 × 150		28.0525	101.8196	25.7981

Figure.3 Block Diagram of the Embedding Stage

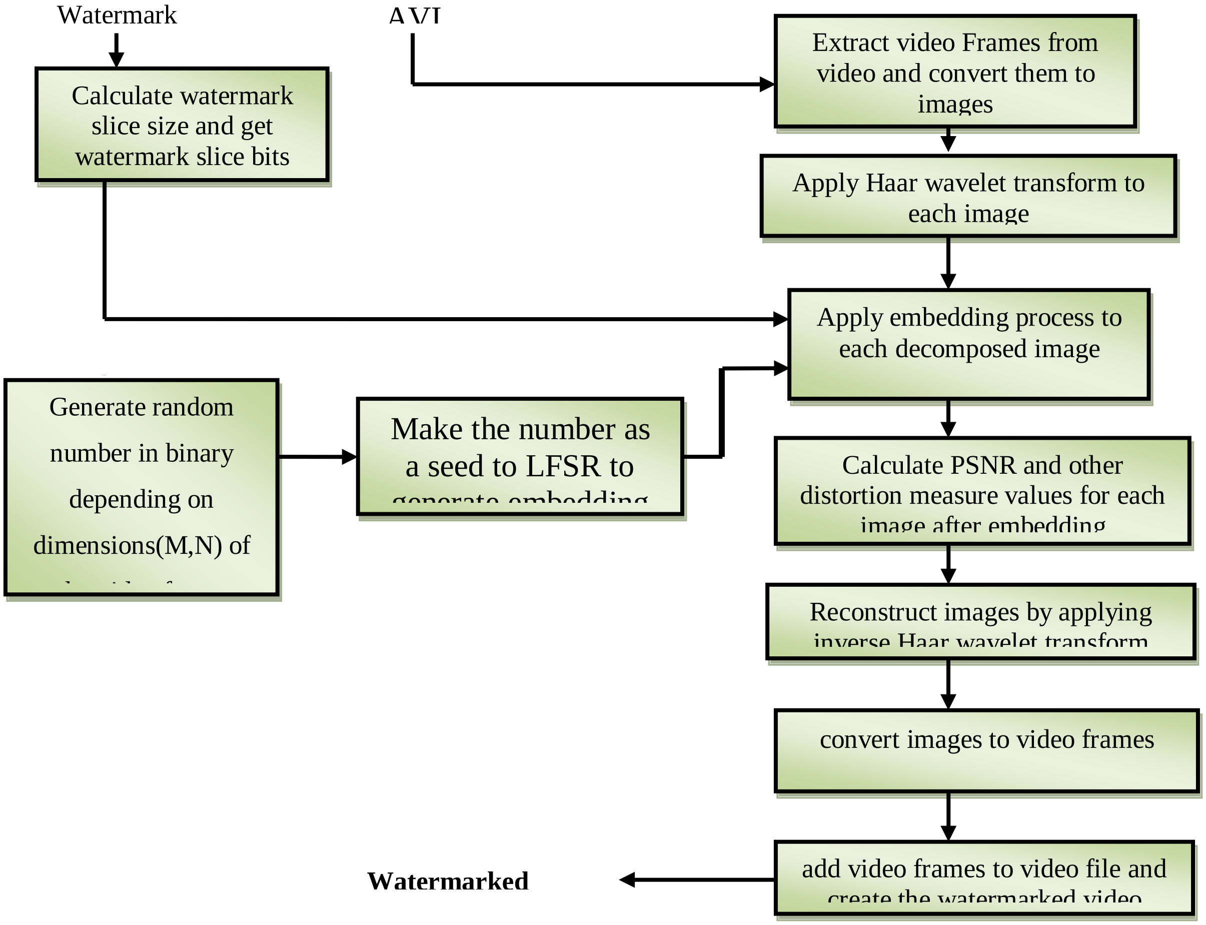

Watermark
AVI
Extract video Frames from video and convert them to images
Calculate watermark slice size and get watermark slice bits
Apply Haar wavelet transform to each image
Apply embedding process to each decomposed image
Generate random number in binary depending on dimensions(M,N) of
Make the number as a seed to LFSR to generate embedding
Calculate PSNR and other distortion measure values for each image after embedding
Reconstruct images by applying inverse Haar wavelet transform
convert images to video frames
add video frames to video file and create the watermarked video
Watermarked

Figure.4 Block Diagram of the Extracting Stage

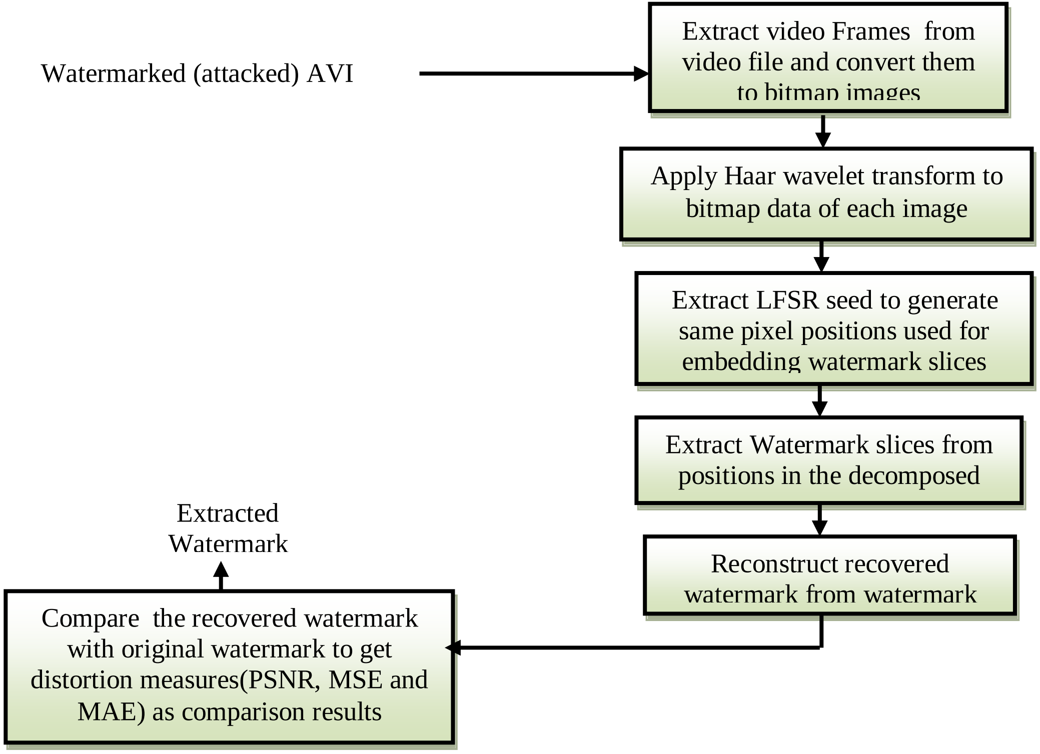

Watermarked (attacked) AVI
Extract video Frames from video file and convert them to bitmap images
Apply Haar wavelet transform to bitmap data of each image
Extract LFSR seed to generate same pixel positions used for embedding watermark slices
Extract Watermark slices from positions in the decomposed
Reconstruct recovered watermark from watermark
Extracted Watermark
Compare the recovered watermark with original watermark to get distortion measures(PSNR, MSE and MAE) as comparison results

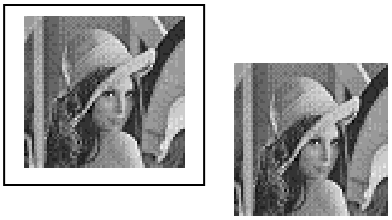
Figure.5 Some of the Frames of Video(Greenland.avi) and Watermark Image(lena.bmp)

Algorithm.1 Embedding Module Algorithm

```
Input: AVI video file, watermark image, watermark information properties.
Output: Watermarked video file.
Step1: Get the AVI video information such as the number of frames in video, frame width, frame height, bit depth and frames per second from the AVI information structure.
Step2: Get the watermark size from the watermark information structure.
Step3: get the first video frame from video file.
Step4: convert the frame to image and store it in a buffer.
Step5:Apply the Haar wavelet transform to the image to get the decomposed image.
Step6: Calculate watermark slice size, then extract the watermark slice bits from watermark buffer according to the watermark slice size and store it in a buffer.
Step7: Generate a random binary number according to the dimension of the (LH) subband of the decomposed image
Step8: Set the state_counter of LFSR to zero (i.e. state_counter=0).
Step9:Send the random binary number as a seed to LFSR, watermark slice size and state_counter to the LFSR state generation function.
Step10:Get the first watermark bit from the watermark slice buffer bits.
Step11:Get a pixel position generated from the LFSR state function in order to embed the watermark bit in this position.
Step12: Embed the watermark slice bit in the LSB of wavelet coefficient of the generated pixel position in the embedding window of (LH) subband of the decomposed image.
Step13: Get the next watermark bit from watermark slice buffer.
Step14: If state_counter<= watermark slice size then goto step 11, else goto next step.
Step15: Calculate the PSNR, MSE and MAE distortion measure values after embedding watermark slice bits and store them in temporary array.
Step16: If the decomposed image corresponds to the first video frame, then embed watermark information which includes watermark size and other information in the decomposed image, else goto next step.
Step17:Embed the LFSR initial (random seed) in the decomposed image.
Step18:Apply inverse Haar wavelet transform to the decomposed image to obtain the bitmap image and store it in a buffer.
Step19:Get the next video frame from video file.
Step20:If frame number<=number of frames in video then goto step 4, else goto step 21.
Step21:apply inverse Haar wavelet transform to all the watermarked images buffers.
Step22:convert the watermarked images buffers to video frames
Step23:Add video frames to video file and create the watermarked video
Step24:End
```

```
Input: Watermarked AVI video
Output: Recovered watermark image buffer
Step1: Get the AVI video information such as number of frames in video, frame width, frame height and
        frames per second from the AVI information structure.
Step2: get the first video frame from video file.
Step3: Read the frame and store it as bitmap image in a buffer.
Step4: Apply the Haar wavelet transform to the image to get the decomposed image.
Step5: If the decomposed image corresponds to the first video frame, then extract the watermark bitmap
        information which includes the size of the embedded watermark with other watermark information,
        else goto  step6.
Step6: Extract the LFSR initial (random seed) which is embedded in the decomposed image.
Step7: If the decomposed image corresponds to the first video frame, then calculate the watermark slice size to
        be extracted from all video frames,  else goto next step.
Step8: Set the LFSR state counter to zero (i.e. state_counter=0).
Step9: Send extracted LFSR seed, watermark slice size and state_counter to the LFSR generator function.
Step10: Initialize the LFSR generator with the initial (seed).
Step11: Get a pixel position generated from LFSR in order to extract watermark slice bits.
Step12: Extract the watermark bit from the LSB of wavelet coefficient in the embedding window according to
        the pixel position and store it in a buffer.
Step13: Increment the LFSR state counter by one  (i.e state_counter=state_counter+1).
Step14: If state counter<=watermark slice size then goto step 13, else goto next step.
Step15: Store the recovered watermark slice bits in a buffer to accumulate all watermark slices.
Step16: Get the next video frame from video file.
Step17: If video frame number <= number of frames in video then goto step 3, else goto next step.
Step18: Reconstruct the recovered watermark image and store it in a buffer.
Step19:End
```

Figure.6 The Original and Extracted Watermark Images



Frame Swapping

We apply Frame Swapping attack on the video sample (traffic.avi). The video consists of 71 frames each has the same format (BMP file) and the dimensions of each frame are (320 × 240) pixels. The watermark image (babon.bmp) is a gray image with size (64*64) pixels.

Table (4) shows the results of distortion measure values of the recovered watermark after applying frame swapping attack to the watermarked video (trafficw.avi) with different numbers of randomly swapped frames.

Salt and Pepper Noise

We apply the salt and pepper noise on the watermarked video (greenland.avi).The video consists of 73 frames each has the same format (BMP file) and the dimensions of each frame are (320*240) pixels. The watermark image (lena.bmp) is a gray image with dimensions (64*64) pixels. the results of distortion measure values of the recovered watermark after applying Salt and Pepper Noise attack with different density values are shown in table (5).

Frame Rotating

We apply the frame rotating attack with degree 180 on the watermarked video (xylophone.avi).The video consists of 141 frames each has the same format (BMP file) and the dimensions of each frame are (320*240) pixels. The watermark image (pepper.bmp) is a gray image with dimensions (64*64) pixels.

The results of distortion measure values of the recovered watermark after applying frame rotating are shown in table (6).

Conclusion

In this paper, a video watermarking scheme has been proposed and implemented for copyright protection of the video file. The proposed system embeds the watermark image in the wavelet domain in decomposed images (frames). The proposed system uses LFSR generator as a random distributor which can spread the effect of embedding over a wide region in the wavelet coefficients by generating random locations in order to embed the watermark bits. Embedding watermark slice bits in the wavelet coefficients of (LH) middle frequency subband will survive various frequency attacks and shows better security of the hidden watermark data. The experimental result shows that the proposed system can recover the watermark with high quality values and resist against different kinds of attacks.

References

Doeerr, G., Dugelay, J.L. 2003. "*A Guide Tour of Video Watermarking*" Signal Processing, *Image Communication,* 18: 263–282.

Pik-Wah, C. 2004. *Digital Video Watermarking Techniques for Secure Multimedia Creation and Delivery*. M. Sc. Thesis, Computer Science and Engineering, The Chinese University of Hong Kong.

Pu, D., Lu, Y., Dai, J. 2010. "*Video Watermarking Approach Based on Temporal difference and Discrete Wavelet Transform*", School of Computer Science and Information technology, Northeast Normal University, Changchun, China.

Essouabi, A., Regragui, F., Ibnelhaj, E. 2009. "*A Wavelet Based Digital Watermarking for Video*". *IJCSIS Int. J. Computer Sci. Information Security,* Vol. 6, No. 1.

Abdulfetah, A.A., Sun, X., Yang, H. 2010. "*Robust Adaptive Video Watermarking Scheme using Visual Models in DWT Domain*". *Res. Article*.

Bhattacharya, S., Chattopadhyay, T., Pal, A. 2006. "*A Survey on Different Video Watermarking Techniques and Comparative Analysis with Reference to H.264/AVC*", IEEE, Kolkata, India.

Chetan, K.R., Raghavendra, K. 2010. "*DWT Based Blind Digital Video Watermarking Scheme for Video Authentication*". *Int. J. Computer Appl.*, Vol. 4, No. 10.

Hassan, M.K. 2006. "*Implementation of Invisible Watermarking Using Multiwavelets*", PH.D. Thesis, Informatics Institute for Postgraduate Studies, Iraqi Commission for Computers and Informatics, Baghdad, Iraq.

Jayamalar, T., Dr. Radha, V. 2010. "*Survey on Digital Video Watermarking Techniques and Attacks on Watermarking*". *Int. J. Engi. Sci. Technol.*, Vol. 2.

George, M., Chouinard, Jean-Yves, Georganas, N. 1999. "*Digital Watermarking of Images and Video using Direct Sequence Spread SpectrumTechniques*". *Proceedings of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering*, Shaw Conference Center, Edmonton, Alberta, Canada.

Al-Kenany, R.K. 1999. "*Stream Cipher Key Generator*", M.Sc. Thesis, Dept. of Computer Science and Information Systems, University Of Technology, Baghdad, Iraq.

Al-Maliky, J.E. 2003. "*Stream Cipher System By Using Linear Feedback Shift Register*", High Diploma Project. Dept. of Computer Science, University Of Technology, Baghdad, Iraq.

Shi, L. 2010. "*Cryptanalysis of a Cascade of Non-Uniforly Clocked Linear Feedback Shift Registers*", M.Sc. Thesis, Dept. of Computer Science and Media Technology, Cjovik University, Norway.

Von, V. 2004. "*On Cryptographic Properties of LFSR-based Pseudorandom Generators*", High Diploma Project, Mannheim University.

How to cite this article:

Hanan Qassim Jaleel and Jane Jaleel Stephan. 2016. Video Watermarking Using Wavelet Transform. *Int.J.Curr.Res.Aca.Rev.*4(3): 1-15.
doi: http://dx.doi.org/10.20546/ijcrar.2016.403.001