Vietnam J Comput Sci (2014) 1:193–206
DOI 10.1007/s40595-014-0021-x

REGULAR PAPER

A robust fingerprint watermark-based authentication scheme in H.264/AVC video

Bac Le · Hung Nguyen · Dat Tran

Received: 3 December 2013 / Accepted: 7 April 2014 / Published online: 29 April 2014

Abstract In this paper, we propose a novel technique that uses fingerprint features with coordinates (x, y), angle and type of feature as watermark information for authentication in H.264/AVC video. We utilize some techniques such as Gabor algorithm, locally adaptive thresholding, and Hilditch's thinning together with heuristic rules and Hamming measurement to optimally extract minutiae vector (x, y, angle, type) from fingerprint as well as to improve accuracy of matching process. Furthermore, to make our scheme robust, the minutiae vector will be converted to binary stream which is increased three times and the lowest frequency of DCT blocks of transition images or frames in H.264 video is properly chosen to hold them. With our proposed technique, the authentication scheme can achieve high capacity and good quality. Experimental results show that our proposed technique is robust against to H.264 encoder, time stretching in video, Gaussian noise, adding blur, frame removal in video, and cutting some regions in the frame of video.

Keywords Video watermarking · H.264/AVC video · Biometric authentication

B. Le (✉)· H. Nguyen
Faculty of Information Technology, University of Science, VNU, Ho Chi Minh City, Vietnam
e-mail: lhbac@fit.hcmus.edu.vn

H. Nguyen
e-mail: kimhung12345@gmail.com

D. Tran
Faculty of Information Sciences and Engineering, University of Canberra, Canberra, ACT 2601, Australia
e-mail: Dat.Tran@canberra.edu.au

1 Introduction

The digital world has invaded many aspects of our lives and moved to all households rapidly in the past decade. More and more digital data are available through various channels such as Internet and media discs. One of the reasons behind the rise of digital data is that users can easily and quickly make a perfect copy of movie, music, or image at large scale with low cost and high quality. Consequently, this has raised concerns about copyright protection against unauthorized duplications and other illegal activities when both content providers and owners realized that the traditional protection methods are no longer efficient and sufficient security [1]. For instance, encryption will not work anymore after decryption since consumers can freely manipulate the decrypted digital content. Other protection methods based on specific header can also easily be broken by removing the header or converting file format. As a result, digital watermarking, the art of hiding copyright information in the robust and invisible manner, has been investigated widely as a perfect complementary technology for copyright protection. With this approach, the embedded data portion considered as evidence to prove copyright of host signal is named watermark. Whereas, the unmarked data portion that needs protected is called host object or unwatermarked object. The marked or watermarked object will be generated after embedding watermark in host object. The relationship among three objects can be demonstrated in Fig. 1a.

Capacity, invisibility and robustness are the most important criteria in a digital watermarking system. *Capacity* is the amount of information (the number of bits) which can be embedded in one unit of the host object (e.g. sample, pixel, scene and so on). *Invisibility* regards to the similarity between unmarked and marked objects. It is usually evaluated by peak signal-to-noise ratio (PSNR). The higher PSRN

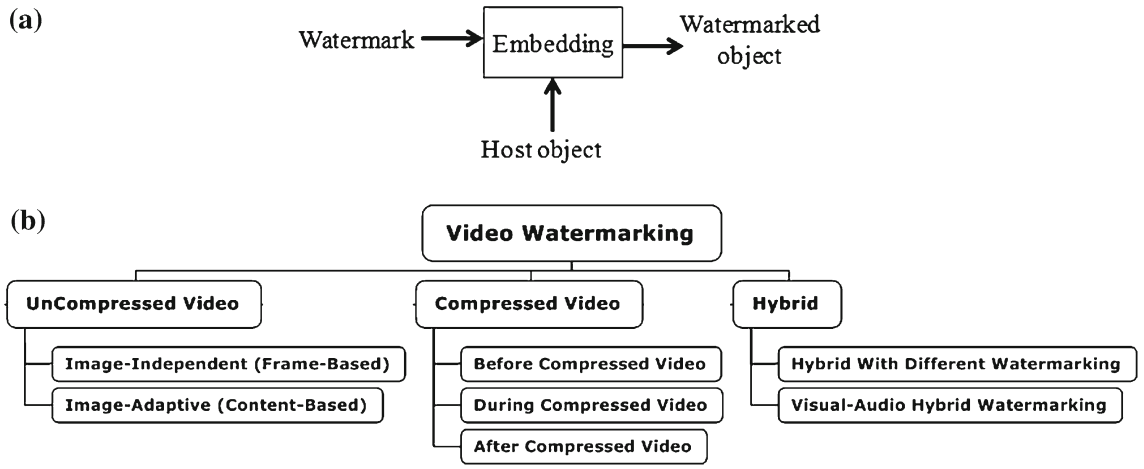


Fig. 1 **a** Digital watermarking system; **b** overview of different types of video watermarking approaches

value gives better invisibility. Finally, *robustness* is considered as the ability of extracting the hidden data from the watermarked signal as well as the survival of the watermark after manipulations or attacks. Because of various operations on digital signal, no watermarking scheme is robust perfectly. As usual, each approach can be robust against to some given and limited alterations. Even though there have been many studies with different approaches, none of the watermarking schemes is strongly enough to meet all requirements at the same time.

The embedded data is usually used to identify the original or copyright information about authors, legal owners, company logo, or signature [2,3]. Recently, biometric information such as iris, face and fingerprint have been utilized and employed as useful watermark [4,5] because it is unique, invariant, and cannot be changed even if stolen. In this paper, we make use important features of fingerprint consisting of the coordinates (x, y), angle and type of features (1 for bifurcation, 0 for ridge ending), namely major minutiae features, as a watermark to authenticate protected content. Hence, there will be about from 30 to 100 minutiae instead of whole fingerprint image embedded in host video [6]. In addition to high reliability of fingerprint, our approach meet the above-mentioned three prerequisites of watermarking problems.

Furthermore, there have been many methods and surveys on digital watermarking [7,8]; however, none of them focuses on video watermarking. Because video protection is not a simple extension of still image protection, more challenges have been encountered. Video watermarking approaches can be classified in Fig. 1b.

Uncompressed video watermarking methods: Most of existing video watermarking methods focus on raw video because of reusability and inheritability from existing image and audio methods. Raw video is simply considered as a sequence of consecutive and equally time-spaced still images. In raw video watermarking algorithm, the inserted code can be casted directly into the video sequence and embedding process can be performed either in the spatial/temporal domain or transformed domain (e.g. DCT, DFT and SVD). Working with uncompressed video allows us to achieve the video-coding format independence and inherit the robustness of image and audio watermarking.

According to how a video is treated, there are two main sub-categories, namely, image-independent and image-adaptive. The first one considers a video as a set of independent still images, so any image watermarking method can be extended to video. Whereas, image-adaptive approaches are based on the video content, therefore, they can exploit more information from the host signal. Different from the first sub-category, content-based watermarking schemes have utilized the concept of Human Visual System (HVS) to adapt more efficiently to the local characteristics of the host signal. These schemes exploit more properties of the image so that they can maximize the watermark robustness while satisfying the transparency requirement.

Compressed video watermarking methods: A video is usually stored in a compression format, such as MPEG-2, MPEG-4 or H.264 to save in the storage space. Probably, raw video is not common because of its large size. Therefore, studies on video watermarking schemes focus on compressed video. The results have shown that inserting watermark into a compressed video allows real-time processing due to low computational complexity. However, it faces problems of video compression standard and payload.

So far, there have been three main approaches dealing with the compressed video watermarking problems shown in Fig. 1b. The first approach embeds watermark into raw video before compressing video such as the H.264/AVC

video watermarking method of Proföck et al. [9] against lossy compression, the strong block selection method against lossy compression standards (e.g. H.264, XviD) of Polyák and Fehér [10] and the new watermarking method based on video 1-D DFT transform and Radon transform of Liu and Zhao [11]. The second approach is to embed watermark directly into the compressed bit stream by changing some parts such as replacing the value of some bytes in the compressed H.264/AVC bitstream [12] and replacing the bits in different blocks based on metadata generated during the pre-analysis [13] in the H.246/AVC compression standard. The third approach allows inserting embedded data into the host compressed video during the encoding such as the watermarking method based on the characteristics of the H.264 standard of Noorkami and Mersereau [14], the hybrid watermark method on the H.264 compression standard used for authentication and copyright protection Qiu et al. [15], the robust watermark method based on H.264/AVC video compression standard of Zhang et al. [16], the watermarking method for the authentication problem on the H.264 video of Su and Chen [7] and the robustness watermarking algorithm on Audio Video Coding Standard (AVS) video of Wanga et al. [17].

Hybrid watermarking methods: Pik-Wah [18] proposed a hybrid approach to improve the performance and robustness of the watermarking scheme. The scene-based watermarking scheme can be improved with two types of hybrid approaches: visual-audio hybrid watermarking and hybrid with different watermarking schemes. The visual-audio hybrid watermarking scheme applies the same watermark into both frames and audio. This approach takes the advantage of watermarking the audio channel, because it provides an independent means for embedding the error-correcting codes, which carry extra information for watermark extraction. Therefore, the scheme is more robust than other schemes which only use video channel alone. The hybrid approach with different watermarking schemes can further be divided into two classes: independent scheme and dependent scheme.

Even though there are many studies with different approaches, none of watermarking schemes is strongly enough capacity, invisibility and robustness at the same time. For instance, the method of Pröfrock et al. [9] against lossy compression H.264/AVC, robustness with regular video attacks and good video quality but not high capacity; the method of Polyák and Fehér [10] gives good results, lower complexity, faster execution, against H.264/AVC and XviD lossy compression process but not robustness with regular video attacks; the method of Liu and Zhao [11] only shows stable to H.264 compression standard, variable geometry and other attacks; and the method of Zou and Bloom [13] is done very quickly at low cost, good compression video quality but not robustness. However, our proposed scheme can achieve high capacity, good quality and robustness. That means our approach can solve three prerequisites of watermarking problems.

The paper is organized as follows: after the Introduction section, all related techniques imployed in this paper will be given in the Sect. 2. The proposed scheme will be demonstrated in Sect. 3. Section 4 will show experimental results and discussion. In final, conclusion as well as future research will be given in Sect. 5.

2 Related works

2.1 Pre-processing fingerprint image

The flowchart of pre-processing fingerprint image can be demonstrated in Fig. 2 with input is a fingerprint image and output is a high quality thinned fingerprint image.

Step 1: filtering

This step will give the high quality of fingerprint image. That means, it makes image clearer, improves the contrast between ridges and valleys, and connects the ridge breaks. There are many methods to enhance the quality of images from simple to complex, from space to frequency domain. However, the implementation of filters over entire image will not be effective. Instead, the filter will be applied on individual block with specific parameters will be more useful [19]. There are four popular context filters, namely, Gabor, Anisotropic, Watson, and STFT, whose parameters depend on the ridge direction and the ridge frequency. Corresponding to fingerprint image and based on experiments, Gabor filter is chosen in this scheme. It is a linear filter and described as follows:

$$G(x, y; \theta, f) = \exp\left\{-\frac{1}{2}\left[\frac{x_\theta^2}{\sigma_x^2} + \frac{y_\theta^2}{\sigma_y^2}\right]\right\} \cos(2\pi f x_\theta),$$

where θ is the orientation of the derived Gabor filter, f is the period of the sinusoidal plane wave, σ_x and σ_y which are standard deviations of the Gaussian envelope along x-axis and y-axis, respectively, and are definite as:

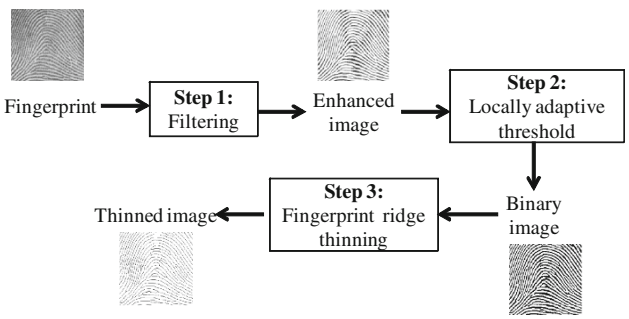


Fig. 2 Flowchart of pre-processing fingerprint

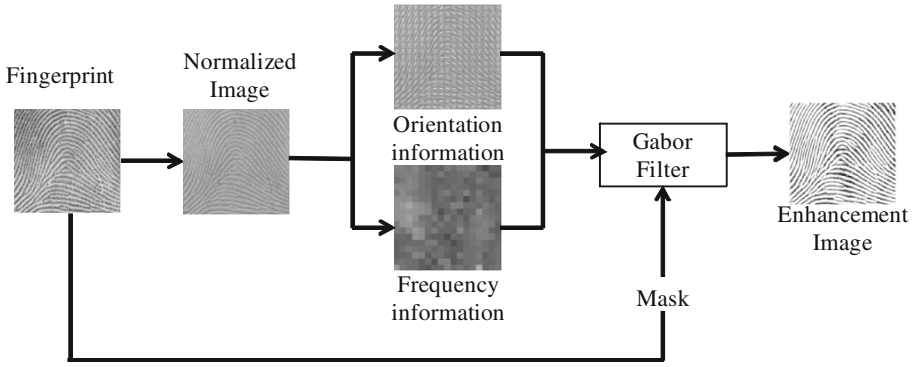


Fig. 3 Apply Gabor filter to fingerprint

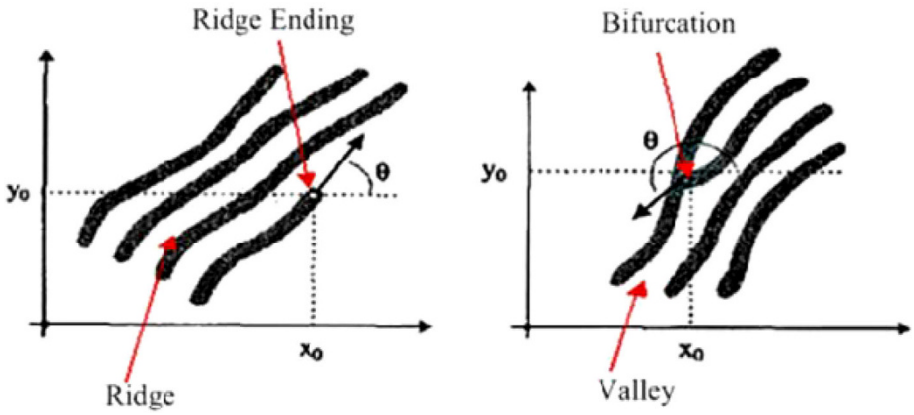


Fig. 4 Ridge ending and bifurcation

$$x_\theta = x\cos\theta + y\sin\theta, \quad y_\theta = -x\sin\theta + y\cos\theta,$$
$$\sigma_x = k_x F(i,j), \quad \sigma_y = k_y F(i,j),$$

To be enhanced by employing Gabor filter, the original fingerprint image is first normalized and then extracts orientation and frequency information for the filtering. The filtering is performed in the spatial domain with a mask (usually sized 17×17). The whole process of enhancing fingerprint image through Gabor filter is described in Fig. 3.

Step 2: locally adaptive thresholding

This step transforms the 8-bit gray scale fingerprint image to 1-bit image with 0-value for ridges (black) and 1-value for valleys (white). It is also called image binarization. The simplest way to get the binary image is based on global threshold T:

$$I'(x,y) = \begin{cases} 1 & I(i,j) > T \\ 0 & I(i,j) \le T \end{cases}.$$

However, this approach is not good in case of fingerprint image. Here, we use local threshold instead. That means the image is first divided into blocks. Within each block, a grayscale pixel will be transformed white if its value is larger than the mean intensity value of the current block.

Step 3: fingerprint ridge thinning

This step will eliminate the redundant pixels of ridges till these ridges are just one pixel wide. Amongst many thinning algorithms such as Holt and Stewart [20], Stentiford [21], Zhang–Suen [22], the experimental results show that Hilditch algorithm [23] is simple algorithm and gives better answer with the fingerprint image. The selected algorithm is described as following:

At point P1 on the ridge, consider the 8-neighbors of pixel P1.Then, calculate A(P1) and B(P1) where A(P1) is the number of pairs (0, 1) in the sequence P2, P3, P4, P5, P6, P7, P8, P9, P2 and B(P1) is the number of neighbor pixels whose values are not zero. Pixel P1 will be transformed from 1 (black) to 0 (white) if it satisfies the following four conditions: (1) $2 \le$ B(P1) ≤ 6; (2) A(P1) $= 1$; (3) P2.P4.P8 $= 0$ or A(P2) != 1; (4) P2.P4.P6 $= 0$ or A(P4) != 1.

2.2 Extracting minutiae feature

There are two types of minutiae: ridge ending and ridge bifurcation are used for extracting and matching shown in Fig. 4. Note that a ridge ending is the point at which a ridge terminates, and a bifurcation is the point at which a single ridge splits into two ridges.

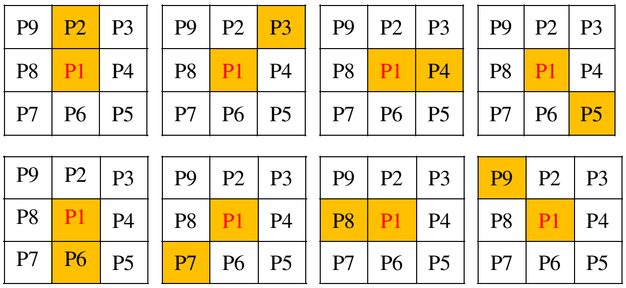


Fig. 5 Cases if P1 is ridge ending

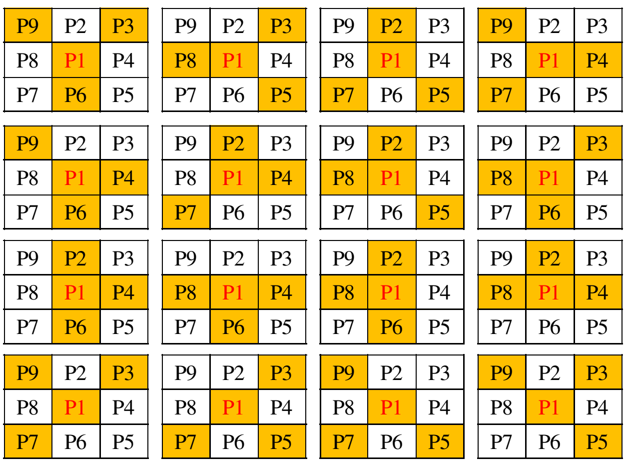


Fig. 6 Cases if P1 is bifurcation

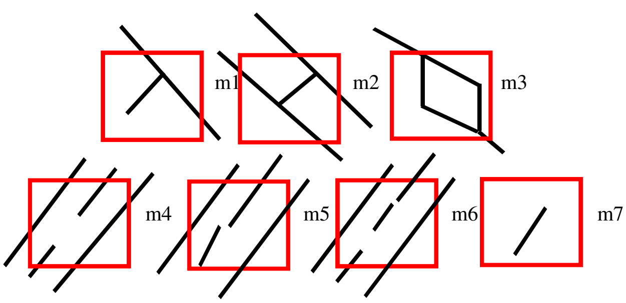


Fig. 7 False minutia structures

By dividing the image into overlapping blocks, sized 3 × 3, central point P1 is considered as ridge ending if it is the following cases (Fig. 5):

Point P1 is bifurcation if it is the following cases (Fig. 6):

The problems of ridge breaks due to lack of or over-ink or over-press will reduce the accuracy of minutiae extraction. There are 7 cases causing such problem considered as following (Fig. 7):

To remove false minutiae, we use heuristic rules as follows:

If the distance between one bifurcation and one termination is less than T ($T = 7$ by default) and the two minutiae are in the same ridge (m1 case). Remove both of them.

If the distance between two bifurcations is less than T and they are in the same ridge, remove the two bifurcations (m2, m3 cases).

If the distance between two ridge endings is less than T and their directions are coincident within a small angle variation. And they meet the condition that no termination is located between the two ridge endings. Then the two terminations are considered as false minutiae derived from a broken ridge and are removed (m4, m5, m6 cases).

If two terminations are located in a short ridge with length less than T, remove the two ridge endings (m7 case).

Where T is the average inter-ridge width representing the average distance between two parallel neighboring ridges. The following picture illustrates the minutiae extraction process (Fig. 8):

Notably, in the above figure, the red circles correspond to bifurcations (type = 1) and the blue circles correspond to the ridge endings (type = 0).

3 Proposed method

From all the research and general knowledge, this paper proposes a robust authentication in H.264 video based on the minutiae (x, y, angle, type) of fingerprint as follows (Fig. 9):

Our authentication scheme using fingerprint watermark consists of three phases as follows:

3.1 Embedding phase

The flowchart of embedding phase can be demonstrated in Fig. 10a.

First, the H.264 video is decoded into raw frames by the H.264 Decoder. Since the transition frames will loose the least data in the H.264 video encoding phase, they are selected from the raw frames. With each transition frame, it is divided into the 8 × 8 non-overlapping blocks. Discrete Cosine Transformation (DCT) will be applied to the set of blocks . In addition, the minutiae vector (x, y, angle, type) generated from fingerprint image after the pre-processing and extracting minutiae will be converted to binary stream (called S). Since binary sequence is much smaller than the transition frame size, we can increase S three times up to SSS. For instance, with minutiae vector (10, 12, 45, 1), we have S = 000010100000110000101101 1 (with 10 = 00001010, 12 = 00001100, 45 = 00101101, 1 = 1) and SSS = 0000101000001100001011011000010100000110000101101100001010000011000010110 11. With the binary sequence SSS, we can embed one bit (S_k) of sequence S into one 8 × 8 block B_k by the following steps [24]:

Step 1: Choose two lowest frequencies from each block called $B1_k$ and $B2_k$. Select one parameter a such that

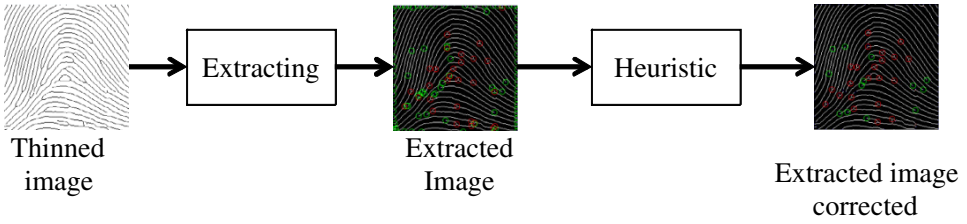


Fig. 8 Minutiae extraction process

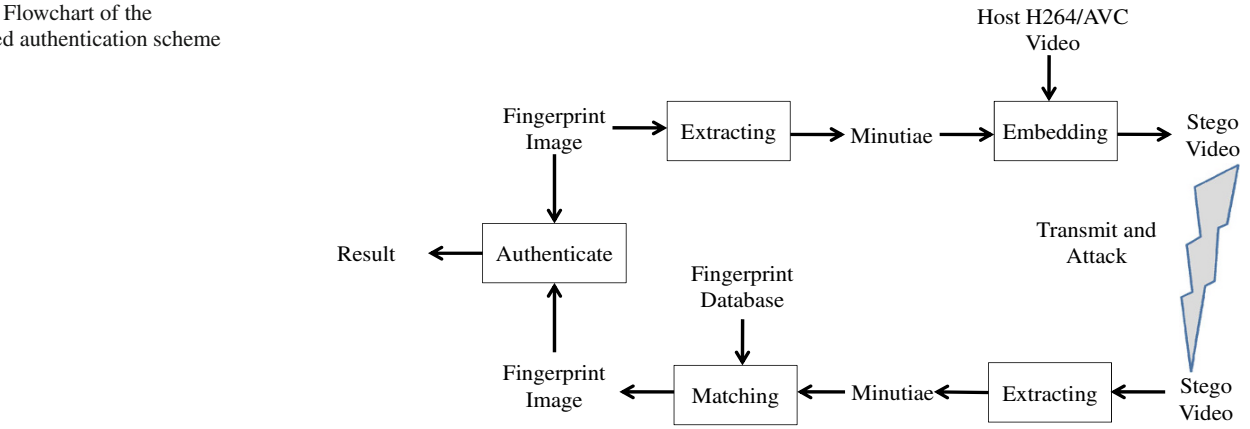


Fig. 9 Flowchart of the proposed authentication scheme

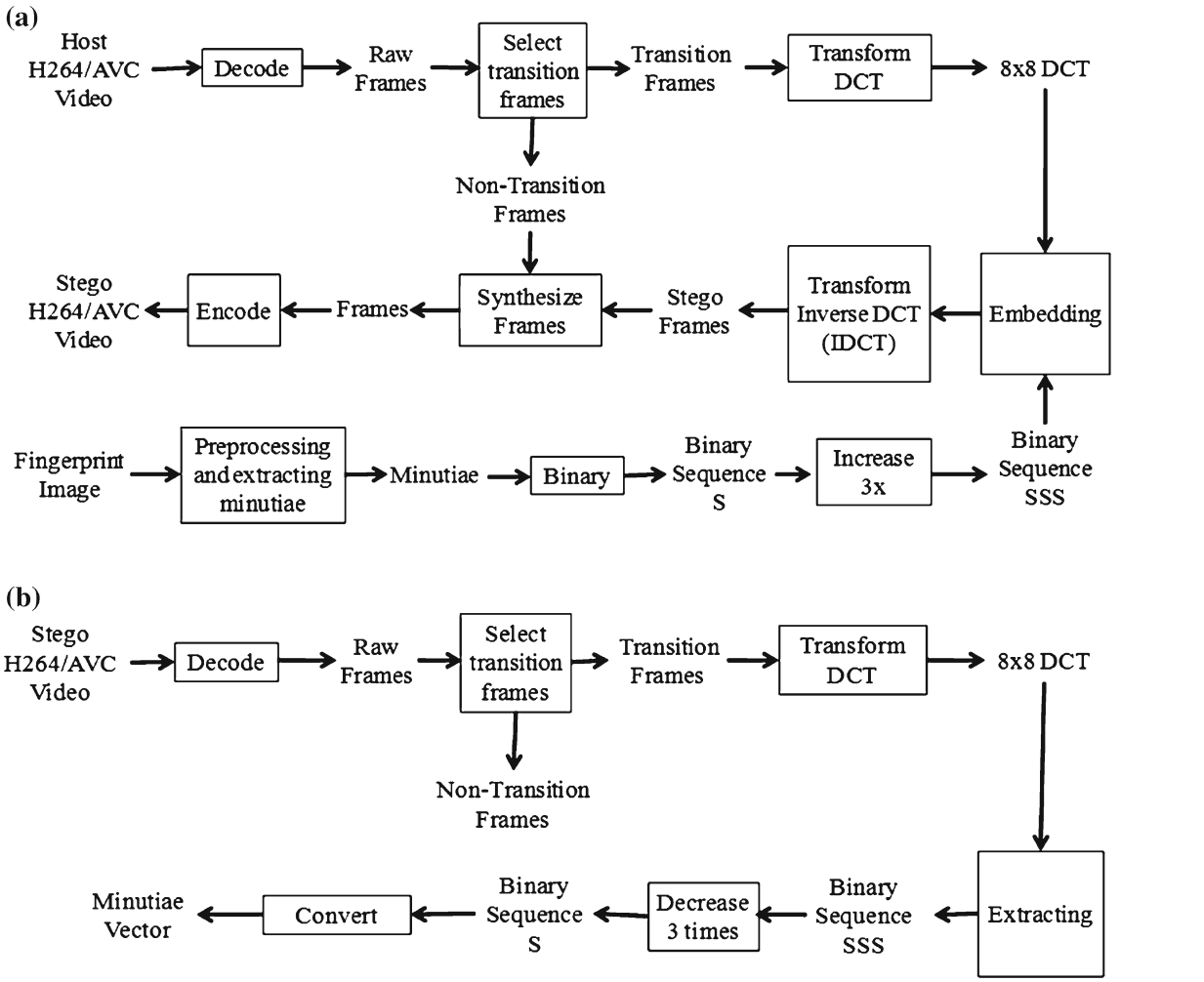


Fig. 10 **a** Flowchart of embedding phase; **b** flowchart of extracting phase

Table 1 The PSNR values of watermarked video

Authenticated Video kid.mp4 (800×480): 2MB 					
Fingerprint Image	**Fingerprint Image Size**	**Size of minutiae vector (bit)**	**PSNR (dB)**	**Size of minutia vector increase 3 times (bit)**	**PSNR (dB)**
1.tif	64.2KB	1152	56.71262	3456	52.02355
1.jpg	13.9KB	1400	55.83754	4200	51.23702
2.jpg	9.41KB	1408	55.95424	4224	49.98475
2.tif	64.5KB	1120	56.68477	3360	50.88121
3.jpg	12.3KB	2208	54.37198	6624	51.31206
3.tif	64.5KB	960	57.42595	2880	51.51483
4.jpg	12.2KB	1856	54.77017	5568	51.68784
4.tif	64.5KB	1312	56.07391	3936	50.09165
6.tif	142KB	846	57.41139	2592	51.50177
10.tif	142KB	416	60.7918	1248	56.60345
Authenticated Video woman.mp4 (320×240): 6MB 					
Fingerprint Image	**Fingerprint Image Size**	**Size of minutiae vector (bit)**	**PSNR (dB)**	**Size of minutia vector increase 3 times (bit)**	**PSNR (dB)**
1.tif	64.2KB	1152	49.66988	3456	44.9808
1.jpg	13.9KB	1400	48.71718	4200	44.1167
2.jpg	9.41KB	1408	50.29945	4224	44.33
2.tif	64.5KB	1120	49.91917	3360	44.1156
3.jpg	12.3KB	2208	45.10393	6624	42.044
3.tif	64.5KB	960	51.66206	2880	45.7509
4.jpg	12.2KB	1856	45.84466	5568	42.7623
4.tif	64.5KB	1312	50.62513	3936	44.6429
6.tif	142KB	846	52.09545	2592	46.1858
10.tif	142KB	416	54.59979	1248	50.4114

$a = 2(2t + 1)$ with t is a positive integer ($0 \leq t \leq 127$) (t = 4, a = 18 by default).

Step 2: Calculate distance between the two frequencies, $d = |B1_k - B2_k| \,(\mathrm{mod}\, a)$.

Step 3: Binary bit S_k will be embedded into frequencies $B1_k$ and $B2_k$ according to the following rules:

- If $S_k =$ '1' and $d \geq 2t+1$, we do not change anything. If $S_k =$ '1' and $d < 2t + 1$, either $B1_k$ or $B2_k$ will be changed such that $max(B1_k, B2_k) = max(B1_k, B2_k) + INT(0.75 \times a) - \mathrm{d}$.
- If $S_k =$ '0' and $d < 2t + 1$, we do not change anything. If $S_k =$ '0' and $d \geq 2t + 1$, either $B1_k$ or $B2_k$ will be changed such that $max(B1_k, B2_k) = max(B1_k, B2_k) + INT(0.25 \times a) - -d$.

The three above steps will be repeated until the minutiae vector *SSS* is completely embedded in transition frames. To obtain the stego frames (the watermarked signal), Inverse Discrete Cosine Transformation (IDCT) will be applied to each block before combining all together. Afterwards the H.264/AVC encoder will be applied to the synthesized frames to obtain stego H.264/AVC video.

3.2 Extracting phase

The watermarked H.264 video may be attacked when it is transferred on a public channel. Therefore, the received H.264 video must be decoded into the raw frames by H.264 decoder. Similar to the embedding phase, the transition frames are selected from the raw frames then are divided into the 8×8 non-overlapping blocks. Discrete Cosine Transformation (DCT) will be applied to the set of blocks before extracting the minutiae vector. According to our approach, each minutia will be taken out based on selecting two lowest frequencies called $B1_k$ and $B2_k$ from each block. Then, based on the distance $d = |B1_k - B2_k| \,(\mathrm{mod}\, a)$, minutia will be conducted as follows: If $d \geq 2t+1$ then $S_k = 1$ and if $d < 2t + 1$ then $S_k = 0$. After extracting, we get the binary sequence *SSS*. To obtain the minutiae vector, we decrease *SSS* three times down to *S*. The whole flowchart of this phase can be described in Fig. 10b.

Table 2 Authentication without attack when embedding into the randomly selected frames

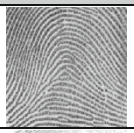
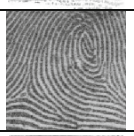
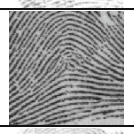
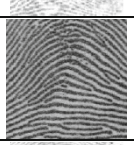
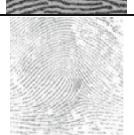
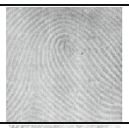
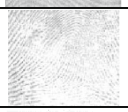
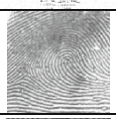
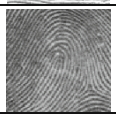
Fingerprint Image Name	Fingerprint Image	Fingerprint Image Size (KB)	Embedded Minutiae Size (bit)	Extracted Minutiae Size (bit)	Bit Error	D	1 - D	Authentication
1.tif		64.2	1152	1152	435	0.377604	0.622396	True
1.jpg		13.9	1400	1400	558	0.396307	0.603693	False
2.jpg		8.89	1408	1408	540	0.383523	0.616477	True
2.tif		64.5	1120	1120	439	0.391964	0.608036	False
3.jpg		12.3	2208	2208	849	0.384511	0.615489	False
3.tif		64.5	960	960	391	0.407292	0.592708	False
4.jpg		12.2	1856	1856	689	0.371228	0.628772	True
4.tif		64.5	1312	1312	489	0.372713	0.627287	True
5.jpg		11.7	1984	1984	377	0.368164	0.631836	True
5.tif		64.5	1024	1024	734	0.36996	0.63004	True
6.jpg		13.6	1664	1664	601	0.361178	0.638822	True
6.tif		142	864	864	315	0.364583	0.635417	True
7.jpg		8.51	1408	1408	535	0.379972	0.620028	True

Table 2 continued

7.tif		64.5	1696	1696	635	0.37441	0.62559	True
8.jpg		11	2304	2304	908	0.394097	0.605903	False
8.tif		142	416	416	158	0.379808	0.620192	True
9.jpg		13.6	1984	1984	750	0.378024	0.621976	True
9.tif		142	512	512	199	0.388672	0.611328	False
10.jpg		9.41	1696	1696	628	0.370283	0.629717	True
10.tif		142	416	416	137	0.329327	0.670673	True
11.jpg		16.6	1504	1504	551	0.366356	0.633644	True
11.tif		64.5	992	992	377	0.38004	0.61996	True

3.3 Matching phase

This phase is to authenticate the legal of host H.264 video by matching the extracted minutiae vector with fingerprint database. Since minutiae vector is considered as a binary stream, Hamming distance is used to achieve good accuracy in authentication. The Hamming distance between two vectors $A = a_1a_2 \ldots a_n$ and $B = b_1b_2 \ldots b_n$ is determined as $D = \frac{1}{n}\sum_{i=1}^{n} |a_i - b_i|$.

If D is less than a preset threshold $D_0 (D_0 = 0.5$ by default) then 2 bit strings are matching. If there are several matching vectors, the smallest value of D is selected.

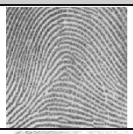
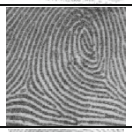
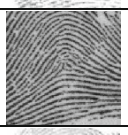
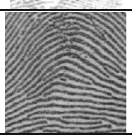
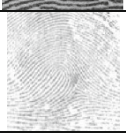
4 Results and discussion

Experiments were conducted on a PC with Intel(R) Core (TM)2 Duo CPU T5800 2.00GHz, RAM 4GB. The operating system is Windows 7 32-bit and our algorithms were programmed in Microsoft Visual C++ 6.0 and Microsoft Visual Studio 2008 with supporting of OpenCV and MediaNet Suite library. To illustrate our scheme, we used the fingerprint database consisting 1500 samples which were provided by Ministry of Public Security of Vietnam (Ho Chi Minh city branch). To demonstrate authentication ability, we used 11 fingerprint images each of which was saved in TIFF and JPEG formats. Details of these 22 files are listed in Table 1 below. The H.264 videos chosen in experiments are kid.mp4 and woman.mp4 sized 2 MB, 6 MB, respectively.

In our experiments, the peak signal-to-noise ratio (PSNR) is used to evaluate the quality of the watermarked frame. A higher PSNR means that the quality of the marked frame is better. The PSNR is defined as $\text{PSNR} = 10 \times \log_{10} \frac{255^2}{\text{MSE}}$ (dB), where MSE is the mean square error between the original frame and the watermarked one. For a host frame with size of $w \times h$, the formula for MSE is defined as

$$\text{MSE} = \frac{1}{w \times h} \sum_{x=1}^{h} \sum_{y=1}^{w} (G_{xy} - G'_{xy})^2 \quad (1)$$

Table 3 Authentication without attack when embedding into the transition frames

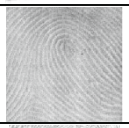
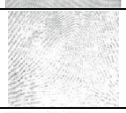
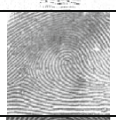
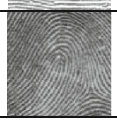
Fingerprint Image Name	Fingerprint Image	Fingerprint Image Size (KB)	Embedded Minutiae Size (bit)	Extracted Minutiae Size (bit)	Bit Error	D	1 - D	Authentication
1.tif		64.2	1152	1152	311	0.269965	0.730035	True
1.jpg		13.9	1400	1400	407	0.289063	0.710937	True
2.jpg		8.89	1408	1408	389	0.276278	0.723722	True
2.tif		64.5	1120	1120	319	0.284821	0.715179	True
3.jpg		12.3	2208	2208	612	0.277174	0.722826	True
3.tif		64.5	960	960	288	0.3	0.7	True
4.jpg		12.2	1856	1856	490	0.264009	0.735991	True
4.tif		64.5	1312	1312	348	0.265244	0.734756	True
5.jpg		11.7	1984	1984	267	0.260742	0.739258	True
5.tif		64.5	1024	1024	521	0.262601	0.737399	True
6.jpg		13.6	1664	1664	422	0.253606	0.746394	True

where G_{xy} and G'_{xy} are the pixel values at position (x, y) of the host frame and the watermarked frame, respectively.

Our proposed scheme obtains good invisibility. Table 1 displays the quality of different videos which are embedded and evaluated by PSNR values.

A frame with $w \times h$ size can be embedded up to $(w \times h)/(8 \times 8)$ bits (each bit is embedded in to a 8×8 block) in the proposed method. If the number of bits to be embedded is bigger than the number of 8×8 blocks, we cannot embed each bit into each block. Instead, we will embed more than

Table 3 continued

6.tif		142	864	864	222	0.256944	0.743056	True
7.jpg		8.51	1408	1408	384	0.272727	0.727273	True
7.tif		64.5	1696	1696	453	0.267099	0.732901	True
8.jpg		11	2304	2304	660	0.286458	0.713542	True
8.tif		142	416	416	114	0.274038	0.725962	True
9.jpg		13.6	1984	1984	537	0.270665	0.729335	True
9.tif		142	512	512	144	0.28125	0.71875	True
10.jpg		9.41	1696	1696	446	0.262972	0.737028	True
10.tif		142	416	416	93	0.223558	0.776442	True
11.jpg		16.6	1504	1504	389	0.258644	0.741356	True
11.tif		64.5	992	992	270	0.272177	0.727823	True

one bit into the lowest coefficients of each block. For instance, after increasing three times, the minutiae of 3.jpg fingerprint image has 6,624 bits and the kid.mp4 video frame size is 800 $\times$ 480. So, we can embed up to $(800 \times 480)/(8 \times 8) = 6{,}000$ bits. In this case, we cannot embed each bit of the minutiae in to each 8×8 block. Therefore, we will embed two bits in the four lowest coefficients of each 8×8 block.

The PSNR in Table 1 is high ($\geq$40 dB). Compared with the results of the PSNR in [11,14,25], the proposed watermarking method is high capacity.

Authentication was considered in the following cases:

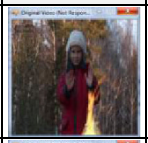
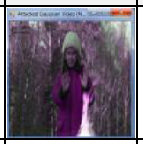
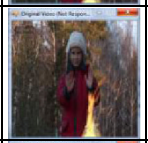
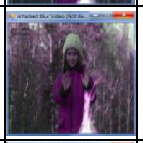
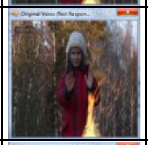
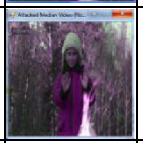
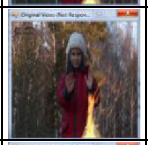
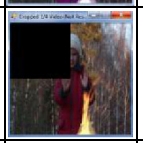
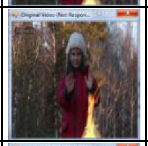
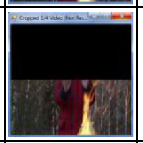
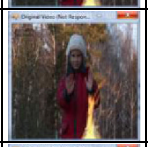
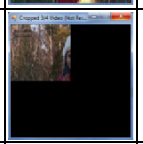
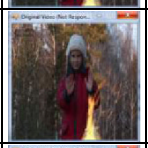
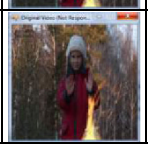
Case 1: There is no attack over public channels. That means the images at both receiver and sender are the same.

Case 2: There are some attacks over public channels. In our scheme, we consider time stretching in video, Gaussian noise, adding blur, frame removal in video, cutting some regions in the frame of video, and converting H.264 video into another video format.

In the first case, the authentication results are recorded in Tables 2 and 3 with the protected video is kid.mp4.

After embedding the minutiae bits into the selected frames, these frames were attacked in the process of H.264/AVC compression such as image subtraction, image convolution, DCT transform, quantization, reconstruction and lossy entropy encoding. With a series of attacks, the experimental results in Tables 2 and 3 are relatively optimistic. Also through the Tables 2 and 3, the selection of

Table 4 Authentication with attack: stretching time, removing frame, Gaussian noise, adding blur, filtering median, cropping image in frame, converting into another video format

Fingerprint Image: 1.tif The embedded minutiae size : 1152 bits The extracted minutiae size: 1152 bits 									
Attacks	**Attacked Video**	**D**	**1 - D**	**Auth**	**Attacks**	**Attacked Video**	**D**	**1 - D**	**Auth**
Stretch time (1s)		0.269965	0.730035	True	Gaussian noise		0.337543	0.662457	True
Stretch time (3s)		0.269965	0.730035	True	Adding blur		0.322691	0.677309	True
Stretch time (5s)		0.269965	0.730035	True	Filtering median		0.329029	0.670971	True
Remove frame (10%)		0.270833	0.729167	True	Crop image (1/4) in frame		0.357566	0.642434	True
Remove frame (25%)		0.270833	0.729167	True	Crop image (1/2) in frame		0.383162	0.616838	True
Remove frame (50%)		0.27691	0.72309	True	Crop image (3/4) in frame		0.426029	0.573971	False
Remove frame (75%)		0.27691	0.72309	True	Convert MP4→ AVI → MP4		0.391267	0.608733	False
Remove frame (90%)		0.27691	0.72309	True	Convert MP4→ WMA→ MP4		0.397621	0.602379	False

"Auth" means Authentication

transition frames is better than the randomly selected frame. The transition frames are proceeded in the Intra prediction, the most content of the transition frame is retained and added in the picture reference list 0 and 1. The Inter prediction uses the picture reference list 0 and 1. Moreover, most of the video frames in the H.264 compression having high homology are in the Inter prediction.

In the second case, we considered some attacks including time stretching in video, Gaussian noise, adding blur, frame removal in video, cutting some regions in the frame of video, and converting H.264 video into another video format. The experimental results are presented in Table 4. The protected video in this case is kid.mp4 and the fingerprint image is 1.tif. The experimental results show that vector extracted has the same size with the one embedded. Based on Hamming values D between the extracted minutiae and the matched sample, the matched sample is always found when threshold D_0 is set to 0.5. However those D values are ranged from 0.25 to 0.43, the authentication is still false in some cases. Moreover, from the results, we can see that our scheme is more robust to attacks. If attack occurs, two minutiae vectors are almost different; therefore, based on values of Hamming distance D, we can recognize if there was an attack.

	Mobile		Football		Table tennis		Foreman		Garden	
	Mark	No mark	Mark	No mark	Mark	No Mark	Mark	No Mark	Mark	No Mark
PSNR	40.7357	–	40.1152	–	40.1590	–	39.8379	–	40.9515	–
Translation	0.7443	0.0185	0.7142	0.0098	0.5769	−0.0084	0.8061	0.0025	0.6580	0.0189
Aspect to 4/3	0.6993	0.0075	0.7118	0.0110	0.6340	−0.0104	0.7314	−0.0029	0.6511	0.0095
to 11/9	–	–	0.7118	0.0110	0.6340	−0.0104	–	–	0.6511	0.0095
to 16/9	0.6993	0.0075	0.7118	0.0110	0.6340	−0.0104	0.7314	−0.0029	0.6511	0.0095
Swap	0.5567	−0.0648	0.5807	0.0472	0.5769	−0.0084	0.7788	−0.0161	0.3984	0.0413
Lost	0.7154	0.0399	0.6988	0.0200	0.6583	−0.0062	0.7772	0.0098	0.6388	0.0048
Gaussian LP	0.4109	−0.0124	0.3814	−0.0693	0.4001	−0.0136	0.4152	−0.0014	0.3750	−0.0692
Level	0.5894	−0.0097	0.6274	0.0176	0.6398	−0.0109	0.7200	−0.0036	0.5816	−0.0134
Rotation 0°	0.6993	−0.0036	0.7096	0.0074	0.6335	0.0013	0.7314	−0.0013	0.6528	−0.0054
1°	0.6923	−0.0072	0.6846	−0.0081	0.614	0.0003	0.7327	−0.0004	0.6551	−0.006
2°	0.6929	−0.0032	0.6872	−0.0015	0.6124	0.0011	0.7300	−0.0002	0.6270	−0.013
3°	0.6893	−0.0079	0.6928	0.0006	0.6132	0.0020	0.7306	0.0001	0.6334	−0.0109
4°	0.6949	−0.0029	0.6910	0.0020	0.6136	0.0012	0.7336	−0.0003	0.6280	−0.0029
5°	0.6937	−0.0021	0.6965	0.0045	0.6107	0.0010	0.7322	0.0004	0.6251	−0.0137
10°	0.6923	−0.0088	0.7060	0.0048	0.6237	0.0013	0.7311	0.0008	0.5961	−0.0026
15°	0.7136	0.0002	0.7217	0.0116	0.6231	0.0046	0.7383	0.0011	0.5899	−0.0202
20°	0.6961	−0.0027	0.7108	0.0056	0.6270	−0.0002	0.7261	0.0006	0.6494	−0.0079
25°	0.6943	−0.0048	0.7183	0.0139	0.6322	0.0012	0.7268	0.0019	0.6418	−0.0102
30°	0.7045	−0.0034	0.7236	0.0012	0.6481	0.0081	0.7139	0.0014	0.6404	−0.0081
35°	0.6837	−0.0054	0.7231	0.0043	0.6416	0.0050	0.7094	0.0013	0.6429	−0.0076
40°	0.6800	0.0042	0.7234	0.0129	0.6296	−0.0001	0.7130	0.0025	0.6417	−0.0096
45°	0.6773	−0.0012	0.7073	0.0026	0.6442	0.0052	0.6869	−0.0011	0.6425	−0.0006

Fig. 11 Extracting results in the paper [11]

Also through Table 4, we see that the authentication model is not affected by stretching time in video. Because this process only affects the time of frame displayed in the screen without changing the frame data, the removing process only affects the authentication model if and only if the transition frames are removed. Depending on the number of removed transition frames, the authentication result will be affected. For instances, there are three transition frames in kid.mp4 video. If we remove 10 % or 25 % frames of the video, it means one transition frame is removed. If we remove 50, 75 or 90 % frames of the video, it means two transition frames are removed. We have also found that the authentication model is robust with other attacks such as Gaussian noise, adding blur, filtering median, cropping image in the frame and not robust with the video format conversion.

Comparing with the experimental results in the papers [14] and [11], our results are also better (Fig. 11; Table 5).

Table 5 Extracting results in the paper [14]

Video sequence	Watermark bits	Re-encoding recovery rate (%)	Bit rate increase (%)
Carphone	44	58	0.80
Claire	22	83	0.44
Mobile	85	85	0.23
Mother	42	68	0.69
Table	38	62	0.31
Tempete	81	83	0.44

5 Conclusion

In this article, we have proposed a video authentication scheme using fingerprint watermark. The experimental results show that our method has not only achieved high capacity together with good quality of watermarked video but also been robust against stretching time, removing frame, Gaussian noise, adding blur, filtering median, and cropping image in the frame attacks. The PSNR values are bigger than 40 dB in most cases. Otherwise, our scheme just embed about 30–100 minutiae, the proposed method is able to provide very high capacity. In the future, we will apply this authentication scheme in other common video standards such as MPEG-2, MPEG-4 and research another measure to replace Hamming distance for improving the accuracy of matching process.

References

1. Ryoichi, S., Hiroshi, Y.: Consideration on copyright and illegal copy countermeasures under IT revolution. Joho Shori Gakkai Kenkyu Hokoku **2001**(52), 37–42 (2001)
2. Ramos, C., Reyes, R.R., Miyatake, M.N., Meana, H.P.: Image authentication scheme based on self-embedding watermarking. Lecture Notes Comput. Sci. **5856**, 1005–1012 (2009)
3. MeenakshiDevi, P., Venkatesan, M., Duraiswamy, K.: A fragile watermarking scheme for image authentication with tamper localization using integer wavelet transform. J. Comput. Sci. **5**(11), 831–837 (2009)
4. Hassanien, E.: Hiding Iris data for authentication of digital images using wavelet theory. Pattern Recognit. Image Anal. **16**, 637–643 (December 2006)
5. Allah, M.M.A.: Embedded biometric data for a secure authentication watermarking. In: IASTED International Conference: Signal Processing, Pattern Recognition, and Applications, pp. 191–196 (2007)
6. Federal Bureau of Investigation (FBI) John Edgar Hoover, The Science of Fingerprints: Classification and Uses, U.S. Government Printing Office, Washington D.C., (2006)
7. Su, P.C., Chen, I. F.: A digital watermarking scheme for authenticating H.264/AVC Compressed Video, 2008 ICS (2009)
8. Potdar, V.M., Han, S., Chang, E.: A Survey of Digital Image Watermarking Techniques. In: IEEE International Conference Industrial Informatics (INDIN), pp. 709–716 (2005)
9. Pröfrock, D., Richter, H., Schlauweg, M., Müller, E.: H.264/AVC video authentication using skipped macroblocks for an erasable watermark. In: Proc. SPIE Visual Communications and Image Processing, vol. 5960, pp. 1480–1489 (2005)
10. Polyák, T., Fehér, G.: Robust Block Selection for Watermarking Video Streams. In: Proceedings of the World Congress on Engineering 2008, WCE 2008, London, U.K., July 2–4, vol. I (2008)
11. Liu, Y., Zhao, J.: A new video watermarking algorithm based on 1D DFT and Radon transform. Signal Process. **90**(2) (2010)
12. Zou, D., Bloom, J.A.: H.264/AVC stream replacement technique for video watermarking. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP (2008)
13. Zou, D., Bloom, J.A.: H.264/AVC Substitution Watermarking: A CAVLC Example, Media Forensics and Security XI. In: Delp, E.J., Dittmann, J., Memon, N.D., Wong, P.W. (eds.) Proceedings of SPIE, vol. 7254 (2009)
14. Noorkami, M., Mersereau, R.M.: Compressed-domain video Watermarking for H.264. In: Proceedings of the International Conference on Image Processing, ICIP, vol. 2, pp. 890–893 (2005)
15. Qiu, G., Marziliano, P., Ho, A., He, D., Sun, Q.: A hybrid Watermarking scheme for H.264/AVC video. In: Proceedings of the 17th International Conference on Pattern Recognition, ICPR, vol. 4, pp. 865–868 (2004)
16. Zhang, J., Ho, A., Qiu, G., Marziliano, P.: Robust video watermarking of H.264/AVC. IEEE Trans. Circuits Syst. II Express Briefs **54**(2), 205–209 (2007)
17. Wanga, Y., Lua, Z., Fana, L., Zheng, Y.: Robust dual watermarking algorithm for AVS video. Signal Process. Image Commun. **24**(4), 333–344 (2009)
18. Chun-Shien, L., Jan-Ru, C., Kuo-Chin, F.: Real-time frame-dependent video Watermarking in VLC domain. Signal Process. **20**(7), 624–642 (2005)
19. Lee, J., Wang, S.D.: Fingerprint feature reduction by principal Gabor basis function. Pattern Recognit. **34**(11), 2245–2248 (2001)
20. Holt, M., Stewart, A.: A parallel thinning algorithm with fine grain subtasking. Parallel Comput. **10**, 329–334 (1989)
21. Stentiford, W.M.: Some new heuristics for thinning binary handprinted characters for OCR. Trans. Syst. Man Cybern. **13**(1), 81–84 (1983)
22. Zhang, T.Y., Suen, C.Y.: A fast parallel algorithm for thinning digital patterns. Commun. ACM **27**(3), 236–239 (1986)
23. http://cgm.cs.mcgill.ca/~godfried/teaching/projects97/azar/skeleton.html
24. Nguyen, X.H., Tran Q.D.: An Image Watermarking Algorithm Using DCT Domain. In: Proceedings of the National Workshop: Selected Issues of Information Technology, pp. 146–151. Science and Technology Publisher, Ha Noi, Vietnam (2005)
25. Shahabuddin, S.; Iqbal, R.; Shirmohammadi, S.; Jiying Z.; Compressed-domain temporal adaptation-resilient watermarking for H.264 video authentication. In: IEEE International Conference on Multimedia and Expo, 2009. ICME 2009, New York (2009)