Signal Processing 90 (2010) 626–639

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

A new video watermarking algorithm based on 1D DFT and Radon transform

Yan Liu, Jiying Zhao *

Multimedia Communications Research Laboratory, School of Information Technology and Engineering, University of Ottawa, 800 King Edward Avenue, Ottawa, Ontario, Canada K1N 6N5

ARTICLE INFO

Article history:
Received 4 November 2008
Received in revised form
29 June 2009
Accepted 5 August 2009
Available online 13 August 2009

Keywords:
Digital video watermarking
1D DFT
Radon transform
Geometric attacks
H.264

ABSTRACT

In this paper, we propose a new video watermarking algorithm based on the 1D DFT (one-dimensional discrete Fourier transform) and Radon transform. The 1D DFT for a video sequence generates an ideal domain, in which the spatial information is still kept and the temporal information is obtained. With detailed analysis and calculation, we choose the frames with highest temporal frequencies to embed the fence-shaped watermark pattern in the Radon transform domain of the selected frames. The adaptive embedding strength for different locations keeps the fidelity of the watermarked video. The performance of the proposed algorithm is evaluated by video compression standard H.264 with three different bit rates; geometric attacks such as rotation, translation, and aspect-ratio changes; and other attacks like frame drop, frame swap, spatial filtering, noise addition, lighting change, and histogram equalization. The main contributions of this paper are the introduction of the 1D DFT along temporal direction for watermarking that enables the robustness against video compression, and the Radon transform-based watermark embedding and extraction that produces the robustness against geometric transformations. One of the most important advantages of this video watermarking algorithm is its simplicity and practicality.

1. Introduction

Digital multimedia data are spreading rapidly through various channels. A perfect copy of a movie, music, and image can be easily obtained with low cost and high quality. Copyright protection is quite necessary against unauthorized duplications and other illegal practices. The traditional protection method—encryption—could not be very efficient since it will not work anymore after decryption. Digital watermarking is a new technology to hide information in host signal in an invisible or inaudible way. A robust digital watermarking works perfectly for copyright protection. Host signal could be any type of multimedia data such as, text, software, image, audio, video, and 3D meshes.

Image watermarking algorithms have to keep image quality and to be robust against general image processing, such as, lossy compression, filtering, noise addition, and geometric transformation. Geometric transformations, such as, rotation, scaling, and translation (RST), are considered the most challenging attacks for image watermarking. There are many watermarking algorithms [1–7] proposed, based on the Fourier–Mellin transform [2], Radon transform [3], feature points [4–6], log-polar mapping and phase information [7], to deal with geometric transformations for image watermarking. We have introduced, analyzed and evaluated these typical RST invariant image watermarking algorithms through implementation, and pointed out their advantages and disadvantages [1].

* Corresponding author.
E-mail address: jyzhao@site.uottawa.ca (J. Zhao).

0165-1684/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2009.08.001

However, this paper will focus on video watermarking. Compared to image watermarking, there are more requirements specific for video watermarking, due to the fact that there are a large amount of data and inherent redundancy between frames in video. A good video watermarking algorithm must be robust against video compression, frame dropping, frame swapping, geometric attacks, etc. Some existing video watermarking algorithms embed and extract watermark in compression domain [8–15]. For example, Zhao et al. [12] proposed a fast video watermarking algorithm by a least modification of the motion vectors in the compressed video bitstream, in which a novel fast estimation of motion vector was presented to make the algorithm effective and practical. The advantage of these algorithms is the possibility for real-time video watermarking. However, one drawback could be that these watermarking algorithms are limited to a specific video compression standard and may not survive video format conversion. Some other video watermarking algorithms use spatial domain [16–18] or other transformed domains, such as, discrete wavelet transform (DWT) and discrete Fourier transform (DFT) [19–21]. For example, Deguillaume et al. [21] proposed a video watermarking algorithm robust to possible attacks based on 3D DFT and 3D log-polar-log mapping. These algorithms are not limited to one specific compression standard. However, due to the computation complexity, it would be harder to implement real-time video watermarking. As related work, Lu and Liao [14] proposed a video watermarking algorithm for MPEG-4 object protection, by using eigenvectors of a video object for synchronization from rotation and flipping. Wang et al. proposed an MPEG video watermarking algorithm robust against geometric attacks [15], in which the geometric attacks are handled with separated 2D DCT, full DCT and block DCT. Three different methods are proposed, respectively, to deal with cropping, downscaling and frame dropping.

In this paper, we propose a new video watermarking algorithm robust to H.264 video compression and RST attacks. Our new video watermarking algorithm is based on the 1D DFT and Radon transform. This algorithm is not processed in compression domain; therefore, it is not limited to any specific compression standard. This algorithm avoids to use 3D video transformation to reduce computation complexities; instead, it uses the 1D DFT along the temporal direction of video. We call this domain temporal frequency domain. In this domain, the spatial information is still kept and the temporal frequency information is obtained. We analyze the temporal frequency information to choose the embedding frames and use the spatial information to decide the watermark embedding strength within the selected frames. The watermark pattern is generated as pseudo-random noise, which makes the watermark detection easy. We adjust the local embedding location and the embedding strength to keep the fidelity of the watermarked video.

The main contributions of this paper are the introduction of the 1D DFT along the temporal direction for watermarking that enables the robustness against video compression, and the Radon transform based watermark embedding and extraction that produces the robustness against geometric transformations. One of the most important advantages of this video watermarking algorithm is its simplicity and practicality.

The rest of the paper is organized as follows. In Section 2, we propose the new video watermarking algorithm based on the 1D DFT along temporal direction and the Radon transform in the temporal frequency domain. In Section 3, we discuss some important implementation strategies. In Section 4, we will present the implementation and experimental results of the proposed algorithm. Finally, we conclude this paper by Section 5.

2. The proposed 1D DFT and Radon transform based video watermarking algorithm

In this section, we will introduce two enabling techniques—the 1D DFT in temporal direction and the Radon transform in temporal frequency domain. Then we will propose the watermark embedding and extraction procedures based on these two important techniques.

The proposed video watermarking algorithm segments video into groups of pictures (GOP); applies the 1D DFT to each GOP to transform the GOP into the temporal frequency domain; then embeds and extracts watermark in the temporal frequency domain by using the Radon transform.

2.1. 1D DFT in temporal direction

The 1D DFT in temporal direction transforms a group of pictures (GOP) into a temporal frequency domain. In this domain, the spatial information and temporal frequency information exist in the same frame. Higher frequencies correspond to the fast motion from one frame to other frames.

The 1D DFT of a video $f(x,y,t)$ of size $M\times N\times T$, in which, $M\times N$ is the size of each frame and T is the total number of frames in the GOP, is shown as follows [22]:

$$F(u,v,\tau)=\sum_{t=0}^{T-1}f(x,y,t)e^{-j2\pi(t\tau/T)} \tag{1}$$

and the corresponding inverse 1D DFT is defined as follows:

$$f(x,y,t)=\frac{1}{T}\sum_{\tau=0}^{T-1}F(u,v,\tau)e^{j2\pi(t\tau/T)} \tag{2}$$

The discrete Fourier transform $F(u,v,\tau)$ also can be expressed alternatively using the exponential form as

$$F(u,v,\tau)=|F(u,v,\tau)|e^{j\phi(u,v,\tau)} \tag{3}$$

$$|F(u,v,\tau)|^2=\mathrm{Re}^2(F(u,v,\tau))+\mathrm{Im}^2(F(u,v,\tau)) \tag{4}$$

$$\phi(u,v,\tau)=\tan^{-1}\left[\frac{\mathrm{Im}(F(u,v,\tau))}{\mathrm{Re}(F(u,v,\tau))}\right] \tag{5}$$

where $|F(u,v,\tau)|$ is the magnitude of the Fourier transform and $\phi(u,v,\tau)$ is the phase angle.

$F(u,v,\tau)$ is a 3D signal with two dimensions of spatial information (u,v) and one dimension of temporal frequency information (τ).

Fig. 1. Three consecutive frames.

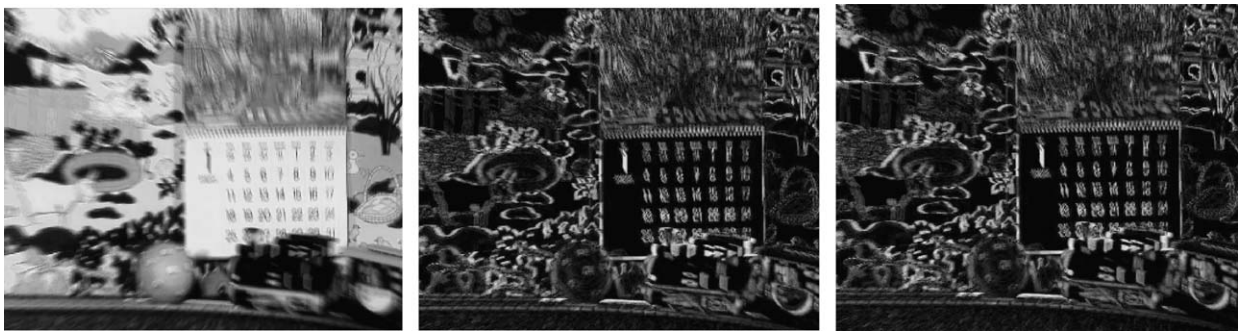

Fig. 2. The 1D DFT along temporal direction of the three consecutive frames in Fig. 1.

Figs. 1 and 2 give an example of three consecutive frames and their 1D temporal frequency response, respectively. In the temporal frequency frames, all the spatial information remains and is cumulated in each frame. The temporal frequency information is shown in each frame to indicate the fast or slow temporal motion. The fast moving parts are much more blurring than the slow moving parts.

2.2. Radon transform

The Radon transform represents an image as a collection of projections along various directions [23]. It is used in areas ranging from seismology to computer vision. Refer to Fig. 3, projections can be computed along any angle γ. In general, the Radon transform of $f(x,y)$ is the integral of f along a straight line parallel to the Y'-axis, which can be expressed by

$$R_\gamma(x') = \int_{-\infty}^{\infty} f(x'\cos\gamma - y'\sin\gamma, x'\sin\gamma + y'\cos\gamma)\,dy' \qquad (6)$$

where

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\gamma & \sin\gamma \\ -\sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

In our case, the vertical projection is used during Radon transform. The 1D projection along the vertical direction

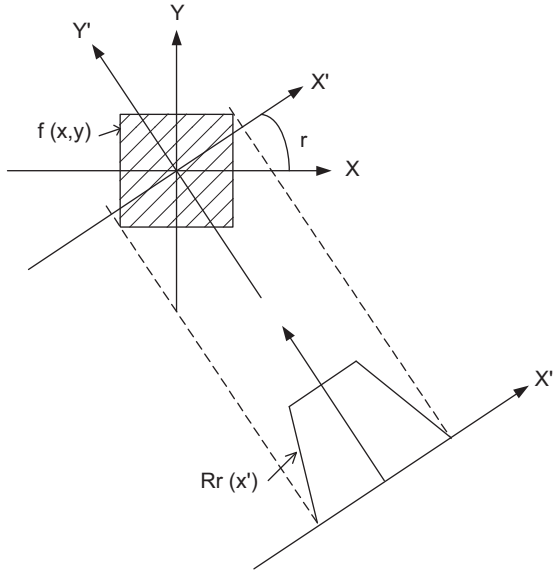

Fig. 3. Radon transform.

to transfer a frame into a 1D array can be expressed as follows:

$$G(u,\tau) = \sum_{v=0}^{N-1} F(u,v,\tau) \qquad (7)$$

where $F(u,v,\tau)$ is the τth frame in the temporal frequency domain, with a size of $M \times N$. $G(u,\tau)$ is the 1D projection of $F(u,v,\tau)$ along the vertical direction (v direction). $G(u,\tau)$ is a 1D array with a length of M.

There have been a number of image watermarking algorithms proposed based on the Radon transform. For example, the algorithm proposed by Lin et al. [3] is based on this principle. They embed the watermark into a one-dimensional (1D) signal by taking the Fourier transform of the image, resampling the Fourier magnitudes into the log-polar coordinates, and then summing a function of those magnitudes along the log-radius axis. For an extensive review on the Radon transform based watermarking algorithms, refer to [1].

2.3. Watermark embedding

Most watermarking systems embed watermark in the luminance component. Considering color video, we embed watermark in the blue channel since the human visual system is less sensitive to this component [24].

In this algorithm, we try to hide a sequence of random numbers into the temporal frequency domain of a video clip. We generate watermark pattern as random positive numbers. It could be any information which are intended to be hidden into the video contents.

We divide a video sequence into groups of pictures (GOP). Then, we apply the 1D DFT to each of GOP to transform it into the temporal frequency domain. In this domain, the spatial information remains and the temporal frequency information is obtained. The DC frame is the one with all spatial information and zero temporal frequency information of the whole GOP. From mathematic point of view, it is just the summation of all the frames in one GOP in the spatial domain. The AC frames refer to the ones which have different temporal frequency information. In our implementation, we choose two symmetrical AC frames furthest from the DC frame as our watermark embedding location. The embedding strength is adaptive to the target video in order to keep the quality of the watermarked video. Depending on the payload, more embedding frames can be considered to meet the requirement.

The watermark embedding procedure consists of the following steps:

(1) Divide the original video into groups of pictures (GOP) with a fixed number of frames.
(2) Compute the 1D DFT along the temporal direction of each GOP. Choose two symmetric AC frames for watermark embedding.
(3) Generate a random sequence of a fixed length by using a pseudo-random generator, which is a spread spectrum consisting of positive values. Here, in our implementation, we generate 32 random numbers.
Then, we spread the generated random numbers into a watermark pattern w by zero-padding in between the generated random numbers. The length of w equals the horizontal width of target video frame.
(4) Apply the Radon transform to the chosen frame (τ) along the vertical direction to get the 1D array $G(u,\tau)$, by using Eq. (7).
(5) Embed the watermark pattern directly into the target frame (τ) by using the proportional embedding method. During implementation, we replace each pixel of the embedding vertical lines of the selected frame by using the following equation:

$$M(u,v,\tau) = \begin{cases} w(u) \cdot \dfrac{\alpha}{G(u,\tau)} \cdot F(u,v,\tau), & w(u) \neq 0 \\ F(u,v,\tau), & w(u) = 0 \end{cases} \quad (8)$$

where $M(u,v,\tau)$ represents the pixel value after watermarking; $F(u,v,\tau)$ is the original value before watermarking in the temporal frequency domain; $w(u)$ is the watermark value at the horizontal position u; and α is the watermark embedding strength.
This proportional embedding method does not change the frequency distribution in the target frame significantly, so it helps to keep the fidelity of the watermarked video. Note we directly embed watermark in the temporal frequency domain ($F(u,v,\tau)$) instead of Radon transform domain ($G(u,\tau)$) by using Eq. (8). However, this embedding method produces an effect of embedding watermark in the Radon transform domain to the watermark extraction process.
(6) Finally apply the inverse 1D DFT with the modified magnitudes and the original phases of the blue channel. Then, convert this channel back together with other two channels to get the watermarked video.

During the embedding process, the key points are the watermark embedding location and strength. The fidelity of the watermarked video depends on the embedding strength. Fig. 4 shows the unwatermarked and watermarked frames in the temporal frequency domain. Note that after the inverse 1D DFT, the watermark is invisible in the watermarked video in the spatial domain. Fig. 5 shows an original frame from video "Mobile" and its corresponding watermarked frame with a PSNR of 40.7357 dB. A watermarked frame with a PSNR around 40 dB looks very close to the original one. The watermark is invisible in spatial domain although it is visible in temporal frequency domain, as shown in Figs. 4 and 5.

2.4. Watermark extraction

To the watermark extraction process, available are the watermarked video that may or may not suffered from attacks, and the watermark key. The procedure of watermark extraction consists of the following steps:

(1) Divide the watermarked video into groups of pictures (GOP) with the same number of frames each as the embedding procedure.
(2) Apply the 1D DFT to each GOP. Select the same frames as embedding procedure for watermark detection.

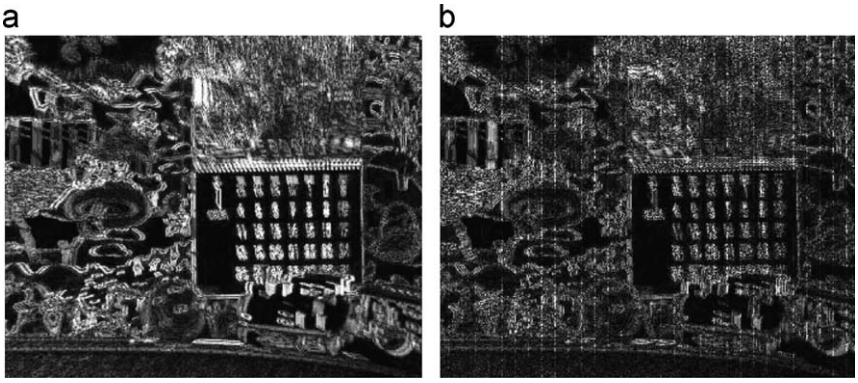

Fig. 4. Original and watermarked frames in temporal frequency domain. (a) Selected original frame from temporal frequency domain. (b) Watermarked version of the frame in (a).

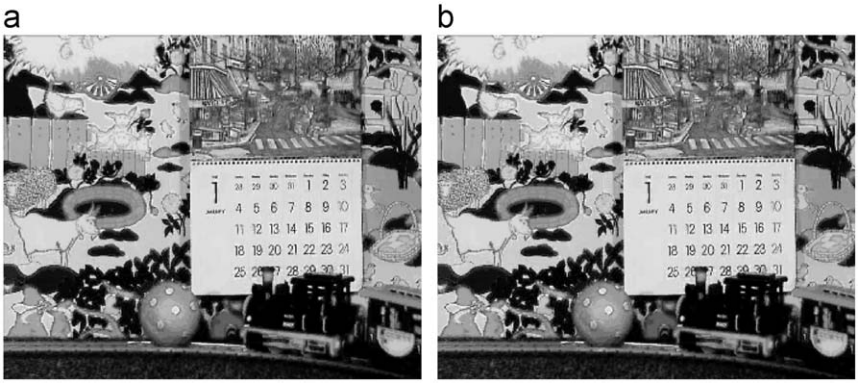

Fig. 5. Original and watermarked frames. (a) First frame of video "Mobile". (b) First frame of watermarked video "Mobile".

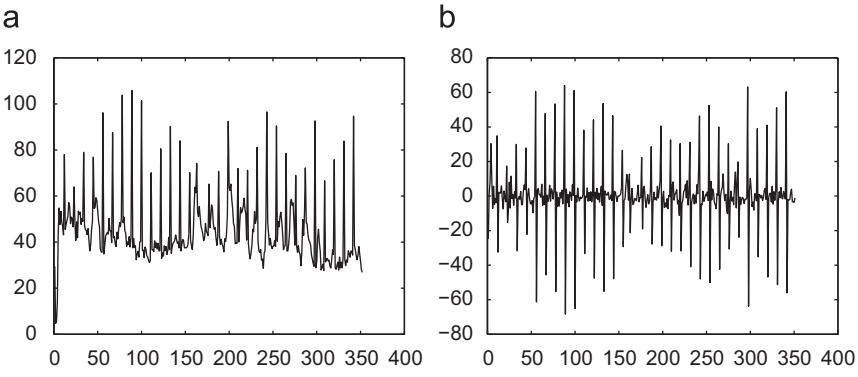

Fig. 6. Radon transform of the watermarked frame in Fig. 4(b) and its gradient.

(3) Apply the Radon transform to this frame (τ) to get an array $G'(u,\tau)$. Fig. 6(a) displays the Radon transform of the watermarked frame in Fig. 4(b). This watermarked frame has not undergone any attack. The watermark location can be detected by the sharp low or high values in the Radon transform domain.

(4) Calculate the gradient of the Radon transform. The gradient of a scalar function $G'(u,\tau)$ with respect to a vector variable $u=(u_1,\ldots,u_n)$ is represented by $\nabla G'$. It is defined to be the vector field whose components are the partial derivatives of G' as follows:

$$\nabla G' = \left(\frac{\partial G'}{\partial u_1},\ldots,\frac{\partial G'}{\partial u_n}\right) \tag{9}$$

where

$$\frac{\partial G'}{\partial u_i} \approx \frac{G'(u_{i+1},\tau)-G'(u_i,\tau)}{u_{i+1}-u_i} \tag{10}$$

Fig. 6(b) shows the gradient of the Radon transform in Fig. 6(a).

(5) Generate the pseudo-random watermark sequence w with the same key as in the embedding process.
(6) Calculate the gradient of the generated watermark sequences by using Eqs. (9) and (10).
(7) Calculate the normalized correlation between the gradient of the Radon transform of the watermarked frame and the gradient of the generated watermark sequence with Eq. (11). If the value of similarity is larger than the threshold, the watermark is successfully extracted. Otherwise, the watermark does not exist or we fail to detect it

$$sim = \frac{\nabla G' \times \nabla w^T}{\sqrt{(\nabla G' \times \nabla G'^T)(\nabla w \times \nabla w^T)}} \tag{11}$$

where $\nabla G'$ and ∇w are, respectively, the gradient of the Radon transform of the watermarked frame and the gradient of the generated watermark sequence.

3. Implementation strategies

In this section, we will explain some important implementation strategies for our video watermarking algorithm.

3.1. Video compression and watermark location selection

Video frames are considered as 3D information, which includes 2D spatial information and 1D temporal information. Therefore, we have to deal with spatial redundancy as well as temporal redundancy during video compression. Video compression is a combination of image compression and motion compensation.

In order to reduce the spacial redundancy in image compression, the DCT and quantization are used, such as, in JPEG and MPEG. The frequency coefficients obtained from the DCT are quantized by dividing a quantization factor in the quantization table to get the quantized coefficients. These quantized coefficients are zeros for most of high frequency coefficients and only a few non-zero coefficients remain for low and middle frequency ones. Therefore, generally speaking, the quantization process is a low-pass filtering process, which removes high frequencies and keeps low frequencies.

To reduce the temporal redundancy, motion estimation and motion compensation are applied in most video compression algorithms. Within a series of frames in one scene, most parts are same except for some motion pixels. Only the changes or differences among those frames are encoded. Generally speaking, to remove temporal redundancy is a high-pass filtering process. Video compression removes still parts which have low temporal frequency and encodes the motion parts which have high temporal frequency.

According to the above analysis, some places that have low or middle spatial frequency and high temporal frequency are the safest locations for watermark embedding against video compression. The 1D DFT along temporal direction could give us a clear way to find the high temporal frequencies. The detailed explanation is given in Section 3.2.

3.2. 1D DFT along temporal direction and watermark embedding

Applying the 1D DFT along temporal direction to a GOP keeps the spatial information and gathers temporal frequency information within one group of frames. The number of frames is the same as that of the original GOP. There is one DC frame that contains no temporal movement. Because of the symmetrical property of the DFT, the left and right frames symmetrical to the DC frame are the same. These frames are named AC frames with non-zero temporal frequency within them. Normally, these AC frames are the target frames for watermark embedding. In order to be able to transform back to get real video frames, symmetric AC frames have to be chosen for watermark embedding. The two symmetrical frames with the highest frequency are the pair of frames farthest from the DC frame if we choose even number of frame as a GOP, and carries the information about fast movement in the temporal direction.

Assume we define that there are T frames in one GOP, where T can be odd or even. $f(x,y,t),\ t \in (0,\ldots,T-1)$ are all frames in the spatial domain in one GOP. $F(u,v,\tau), \tau \in (0,\ldots,T-1)$ are all frames in temporal frequency domain in the same GOP. $F(u,v,0)$ represents the DC frame in the temporal frequency domain. $G(u,\tau)$ is the Radon transform (1D projection along vertical direction) of the selected frame in the temporal frequency domain for watermark embedding. α is the watermark embedding strength. We choose two symmetrical frames from the AC frames. According to our calculation, the relationship between the watermarked frame and the original frame in the temporal frequency domain can be explained as follows:

- If embed watermark in frame pair $F(u,v,1)$ and $F(u,v,T-1)$:

$$f(x,y,t)' = f(x,y,t)\left(1 + \frac{\alpha}{T \cdot G(u,\tau)}(w \cdot e^{j2\pi(1/T)t} + w \cdot e^{j2\pi((T-1)/T)t})\right) \tag{12}$$

- If embed watermark in frame pair $F(u,v,2)$ and $F(u,v,T-2)$:

$$f(x,y,t)' = f(x,y,t)\left(1 + \frac{\alpha}{T \cdot G(u,\tau)}(w \cdot e^{j2\pi(2/T)t} + w \cdot e^{j2\pi((T-2)/T)t})\right) \tag{13}$$

- $\cdots$
- If embed watermark in frame pair $F(u,v,(T-1)/2)$ and $F(u,v,(T+1)/2)$ and when T is odd:

$$f(x,y,t)' = f(x,y,t)\left(1 + \frac{\alpha}{T \cdot G(u,\tau)}(w \cdot e^{j2\pi((T-1)/2T)t} + w \cdot e^{j2\pi((T+1)/2T)t})\right) \tag{14}$$

or, if embed watermark in frame $F(u,v,T/2+1)$ and when T is even:

$$f(x,y,t)' = f(x,y,t)\left(1+\frac{\alpha}{T\cdot G(u,\tau)}(w\cdot e^{j2\pi((T+2)/2T)t})\right) \tag{15}$$

where $f(x,y,t)$ and $f(x,y,t)'$ are, respectively, the original and watermarked frames in spatial domain, and w is the watermark pattern.

According to above equations, no matter which AC frames are used to embed watermark pattern, the first frame in each GOP (I frame) will be modified the most after watermark embedding, shown in the following equation:

$$f(x,y,0)' = f(x,y,0)\left(1+\frac{a\cdot\alpha\cdot w}{T\cdot G(u,\tau)}\right) \tag{16}$$

where a is 1 when one frame in the temporal frequency domain is used for embedding; and a is 2 when two symmetrical frames in the temporal frequency domain are used for embedding. All other frames in the GOP will be also affected but the modification will be smaller.

3.3. Robustness to RST attacks

As we mentioned in Section 3.1, video compression is one of most important attacks to a video watermarking algorithm. Similar to compression, noise addition, filtering, and frame dropping are the attacks, which will not change the watermark locations, but will weaken the watermark strength. For these kinds of attacks, we do not need to conduct re-synchronization. However, there are some other attacks, which do distort the watermark locations in a linear or non-linear way. These attacks include geometrical attack, such as, rotation, scaling, translation (RST). To deal with these attacks, re-synchronization is necessary.

Rotation, scaling and translation attacks are the most challenging attacks for image watermarking algorithms. In case of video, rotation could happen but in very slight angles; scaling could be explained as frame aspect ratio changes; translation is the pixel translation of each frame.

To deal with these attacks, there are different methods [2,3,7,25] for image watermarking. For video watermarking, we also want to re-synchronize the RST attacked video to find the RST parameters or some other RST related properties. The Radon transform in the vertical direction of the target frame has tight relationship with geometrical transformation, such as, rotation, scaling and translation. The proposed video watermarking algorithm utilizes these properties to deal with RST attacks. The gradient of the Radon transform during the watermark extraction enhances the stability of similarity measurement.

For rotation, the rotation angle is also the angle between the Radon transform of the rotated frame and the original frame. Rotation for a video clip is always very slight, probably smaller than 5°. Fig. 7(a) shows an original frame from a video slightly rotated with 5°. Fig. 7(b) is the watermarked frame in the 1D DFT domain of a rotated video corresponding to the frame shown in Fig. 7(a). It is reasonable to exhaustively rotate it back with 1° or smaller each step depending on applications. After each rotation-back, we calculate the similarity between the gradient of the Radon transform of the watermarked frame and the original watermark to search if the watermark pattern exists.

Scaling for video can be frame aspect ratio changes. After frame aspect ratio changes, the watermark location is moved as well. To re-synchronize is always a challenging problem. In our algorithm, the Radon transform and its gradient of the watermarked frame will stretch or shrink as well according to the aspect ratio change. Scaling in vertical direction will not destroy the synchronization of watermark, although it may change the watermark strength. However, scaling in horizontal direction will desynchronize the watermark. In the proposed scheme, the gradient of the Radon transform of the watermarked frame, mentioned in Section 2.4, indicates clearly the location of the watermark pattern. Fig. 8(a) shows a frame within a video clip with frame aspect ratio changed to 16:9 from the original 11:9. The frame with aspect-ratio change is obtained by interpolation. Fig. 8(b) shows the watermarked frame extracted from the video corresponding to Fig. 8(a). Fig. 9(a) presents the Radon transform of the watermarked frame in Fig. 8(b); while Fig. 9(b) shows the gradient of the Radon transform in Fig. 9(a). Each peak in the vector of the gradient indicates the location of a watermark data. After extracting the watermark pattern at the peak locations, we calculate the similarity to evaluate the possibility of watermark existence. Therefore, the proposed algorithm could self-synchronize after aspect ratio changes.

Pixel translation for video frames is another attack to destroy synchronization for watermark location. The starting point for watermark pattern will be shifted due to the translation of a video clip. Fig. 10(a) shows a frame from a translated video, which has been translated 50 pixels to the right. Fig. 10(b) is the watermarked frame in the temporal frequency domain. The start point of watermark sequence has been shifted in the Radon transform. As mentioned before, there are always sharp values which are our watermark values. We can use the gradient of the Radon transform to find out the locations of the watermark values. After picking up the watermark values, we need to find out the start point, in other words, to do the re-synchronization. Fig. 11(a) shows the retrieved watermark sequence from the Radon transform of the watermarked frame. Fig. 11(b) shows the original watermark sequence. In this case, we use filter to locate the watermark position. Many traditional filters, such as, the classical matched filter, amplitude-only filter, inverse filter, phase-only filter and binary phase-only filter can be chosen. However, none of them works well for our matching. Only the phase-only filtering method, which has been proposed in our previous paper [1,26], could give us a sharp peak to indicate the matching value. The filtering process is described as follows:

$$R = IFFT[F_{\phi}(u,v)\cdot G^{*}_{\Phi}(u,v)] \tag{17}$$

where

$$F_{\Phi}(u,v) = e^{-j\Phi_F(u,v)} \tag{18}$$

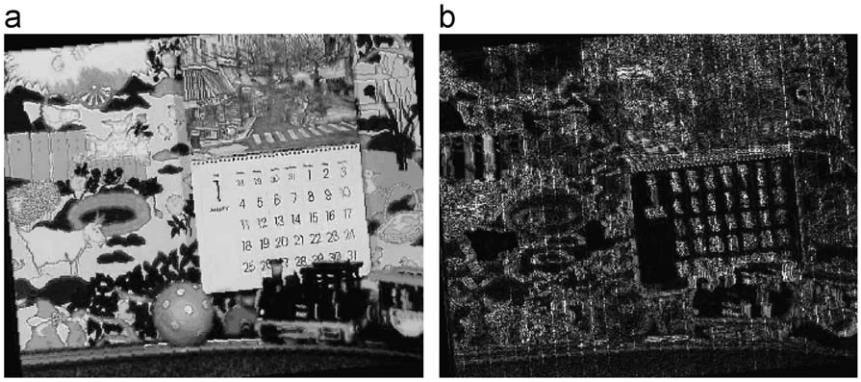

Fig. 7. Original and watermarked frames in a rotated video. (a) A video frame in spatial domain rotated 5°. (b) Watermarked video frame in 1D DFT domain corresponding to (a).

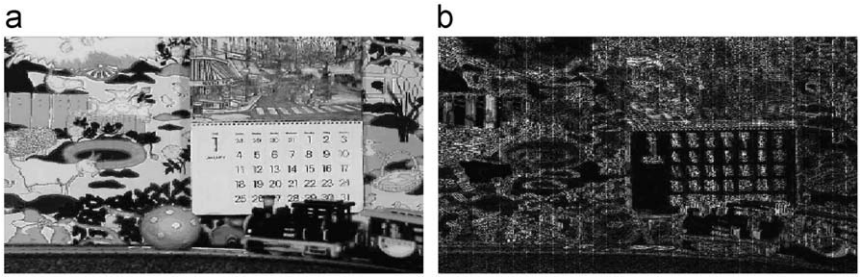

Fig. 8. Original and watermarked frames in an aspect-ratio-changed video. (a) A video frame in spatial domain with aspect ratio changed to 16:9 from 11:9. (b) Watermarked video frame in 1D DFT domain corresponding to (a).

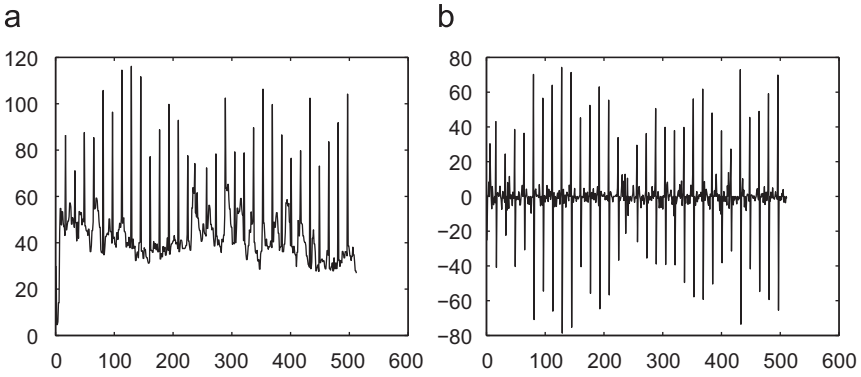

Fig. 9. Radon transform of the watermarked frame after frame aspect ratio changes (a) and its gradient (b).

Fig. 11(c) presents the matching results between the extracted watermark value and the original watermark pattern.

After re-synchronization, the similarity calculation between the retrieved and original watermark array could show if the watermark exists or not.

3.4. Minimum requirement for the target video

According to video compression analysis in Section 3.1, we embed watermark in the high frequencies of the temporal frequency domain. If we modify too much information in this domain, it will degrade the quality of the watermarked video. There is a trade-off between watermark embedding strength and the quality of the watermarked video. The watermark embedding strength depends on the pixel values in the high frequency frame in the temporal frequency domain. In other words, it depends on the amount of moving portions existing in the target video. The more the moving portions in the target video, the higher the watermarking strength could be. Therefore, if in one GOP, all the frames are the same without any change, no watermark could be embedded.

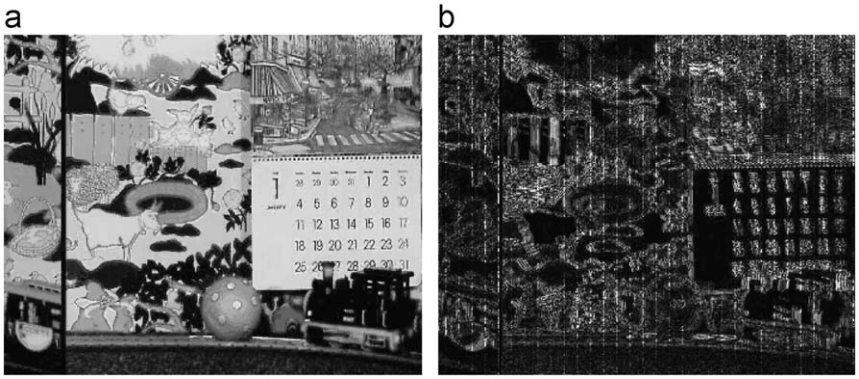

Fig. 10. Original and watermarked frames in a translated video. (a) A video frame in spatial domain translated 50 pixels to the right. (b) Watermarked video frame in 1D DFT domain corresponding to (a).

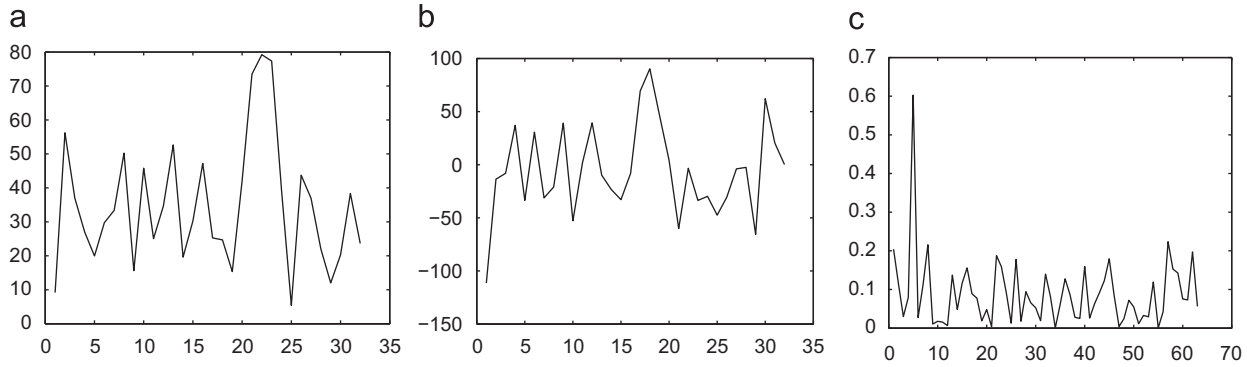

Fig. 11. Re-synchronization of the watermark sequence for the translated video. (a) Extracted watermark sequence. (b) Original watermark sequence. (c) Correlation between (a) and (b) by using phase-only filtering method.

In our algorithm, we automatically control the embedding strength in each GOP according to the pixel values of the high frequency frame in the temporal frequency domain. As the result, the PSNR of all the watermarked videos have been controlled to be around 40 dB, which is considered to be of very good fidelity.

4. Experimental results and evaluation

In this section, we will illustrate the performance of the proposed algorithm. The target videos we tested for our algorithm are shown in Fig. 12. We use five test videos, and their sizes are shown in Table 1.

4.1. Fidelity

We use PSNR as an objective method to check the fidelity of the watermarked video. We compute PSNR for blue channel of each frame between the original video and the watermarked video and average all the values to get the objective PSNR. The PSNR for each watermarked video is shown in Table 2. Human eyes could not recognize the difference between the original video and the watermarked video under these PSNRs.

4.2. Rotation with cropping

Normally, the rotation is very slight for video signals. The rotation angles are not more than 5°. Table 2 shows the experimental results for rotated watermarked and unwatermarked videos. In this table, "Mark" means the results for watermarked videos and "No mark" means the results for unwatermarked videos. In this table, we show the rotated angles up to 45°. The similarity values for watermarked video are much higher than the one for unwatermarked video. Table 2 suggests that the proposed algorithm is very robust to rotation.

4.3. Translation

Translation is another attack which needs re-synchronization. The results for translation are shown in Table 2. No matter how many pixels the translations happened to the watermarked video are, the similarity results for

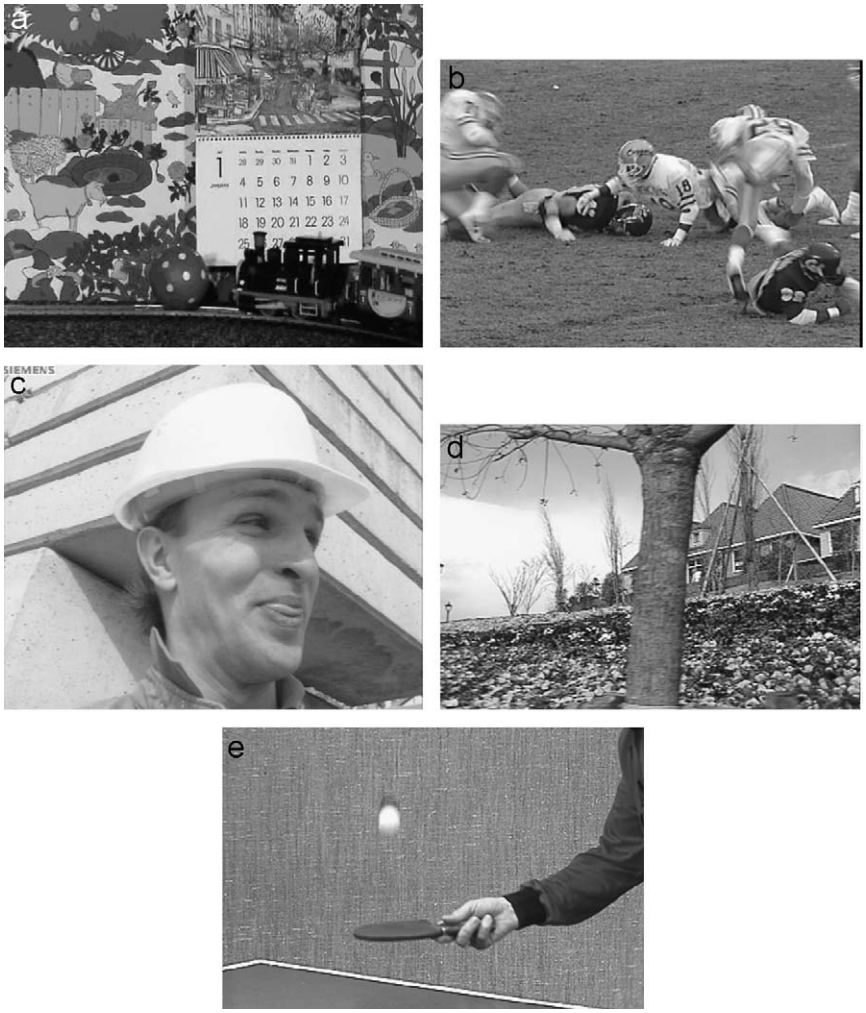

Fig. 12. Test videos. (a) Mobile. (b) Football. (c) Foreman. (d) Garden. (e) Table tennis.

watermarked video are the same after re-synchronization. The phase-only filtering method can correctly find out the translation parameters.

4.4. Frame aspect ratio changes

Frame aspect ratio changes convert the size of the target video signal, similar to scaling for image processing. Here, we consider three popular frame aspect ratios, $\frac{4}{3}$, $\frac{11}{9}$, and $\frac{16}{9}$. Different aspect ratios have been applied to the target video signals than their original ratio. The results are also shown in Table 2. In Table 2, "Aspect to X" means changing the original aspect ratio to X. The similarities for watermarked video are always much higher than the ones for unwatermarked video.

4.5. Frame swapping

Frame swapping means switching the order of frames randomly within one GOP. However, too many frame swaps will degrade video quality. Therefore, we swap frames only once during our experiments. Within one GOP, there is not significant changes among pictures. One swap does not bring too much difference in temporal frequency domain. Our algorithm is robust against the frame swapping according to the experimental results in row "Swap" of Table 2.

4.6. Frame lost

In this experiment, we drop one frame, and borrow one frame from the next GOP. As shown in row "Lost" in Table 2, we can clearly differentiate the watermarked video and the unwatermarked video.

4.7. Spatial filtering

As shown in row "Gaussian LP" of Table 2, our algorithm is also robust to Gaussian low pass filtering process.

Table 1
Target videos and their sizes.

Five videos	Mobile	Football	Table tennis	Foreman	Garden
Size	352 × 288 (CIF)	352 × 240 (SIF)	352 × 240 (SIF)	352 × 288 (CIF)	352 × 240 (SIF)

Table 2
Experimental results for target videos.

	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

4.8. Gray level transformation

By using the gray level $(r+g+b)/3$ to replace the blue channel of the watermarked video, we found that the proposed algorithm is robust to the gray level transformation. The experimental results are shown in row "Level" of Table 2.

4.9. H.264 compression

Tables 3 and 4 show the simulation results for H.264 compression. For each watermarked video, we run the H.264 compression for three times in different bit rates (shown as three columns under each test video). When the compressed video has a PSNR of around 40 dB, which is the acceptable quality for the compressed video, the similarity for watermarked video is between 0.43 and 0.57 with an average of 0.51, and the similarity for unwatermarked video is very low. The higher the PSNR, the higher the similarity value for the watermarked video. Even with lower PSNRs of around 35 dB, we still could successfully separate the watermarked and the unwatermarked videos. Therefore, considering the compression efficiency of H.264, this algorithm is quite robust to H.264 compression.

The proposed video watermarking algorithm is also robust to other video compressions such as, "Indeo3", "Indeo5", "Cinepak" and MPEG2.

4.10. Combination attacks

It is difficult for video watermarking algorithms to resist camera-capturing attack which can be considered as the combination of noise addition, RST transform, lighting change, compression, etc. In this section, to simulate camera-capturing attack, we apply the following combined attacks—H.264 compression, salt & pepper noise, rotation, aspect ratio change, translation, lighting change, histogram equalization—to the target videos. The simulation results are shown in Tables 5 and 6.

In the simulation, the bit rates for H.264 compression are the ones in the middle column under each test video in Tables 3 and 4. The density of the additive salt & pepper noise is 0.001. Rotation angles are from 0° to 45°. The aspect ratio changes are from the original to $\frac{16}{9}$ for each target video. Translations are 20 pixels to the right. The histogram equalization is applied to luminance of target

Table 3
H.264 compression results I.

	Mobile			Football			Table tennis		
PSNR Y	33.59	38.46	48.66	34.48	39.65	48.19	35.24	40.49	48.79
PSNR U	37.93	40.30	47.29	37.87	41.13	47.12	40.21	42.29	47.75
PSNR V	37.68	40.12	47.28	38.71	41.27	47.14	39.60	42.29	48.06
Bitrate (kb)	3119.76	5576.45	12725.85	1837.12	3866.77	8811.44	1376.0	3019.84	7647.12
Mark	0.3446	0.5490	0.6031	0.3890	0.5275	0.5933	0.2645	0.5305	0.6226
No mark	0.0452	0.0408	0.0209	0.0541	0.0385	0.0258	0.0433	0.0234	−0.0334

Table 4
H.264 compression results II.

	Foreman			Garden		
PSNR Y	39.92	43.67	49.70	34.50	39.62	48.00
PSNR U	42.86	45.67	49.37	36.05	40.06	47.32
PSNR V	45.20	47.07	49.88	36.70	39.86	47.12
Bitrate (kb)	1040.80	2444.85	5999.31	1974.36	3774.29	8829.84
Mark	0.4003	0.5749	0.6522	0.3089	0.4300	0.5373
No mark	0.0301	0.0426	0.0195	0.0142	0.0136	−0.0020

Table 5
Combination attacks results I with lighting change.

	Mobile		Football		Table tennis		Foreman		Garden	
	Mark	No mark	Mark	No mark	Mark	No Mark	Mark	No Mark	Mark	No Mark
Rotation 0°	0.5399	−0.0201	0.5110	0.0445	0.5131	0.0885	0.5310	0.0369	0.4289	0.0458
1°	0.5279	−0.0243	0.4623	−0.0072	0.4930	0.0811	0.5241	0.0353	0.4048	0.0366
2°	0.5291	−0.0176	0.4692	0.0221	0.4989	0.0790	0.5234	0.0343	0.4180	0.0491
3°	0.5263	−0.0163	0.4875	0.0310	0.5098	0.0911	0.5229	0.0358	0.4270	0.0519
4°	0.5309	−0.0133	0.5000	0.0346	0.5041	0.0971	0.5227	0.0338	0.4146	0.0481
5°	0.5313	−0.0184	0.4825	0.0293	0.5162	0.0883	0.5306	0.0388	0.4001	0.0301
10°	0.5197	−0.0291	0.4599	0.0131	0.5035	0.0795	0.5213	0.0415	0.3953	0.0466
15°	0.5161	0.0032	0.4993	0.0253	0.4687	0.0637	0.5137	0.0433	0.3995	0.0215
20°	0.4798	−0.0090	0.4974	0.0288	0.4445	0.0536	0.5155	0.0444	0.3947	0.0274
25°	0.4562	−0.0139	0.5284	0.0455	0.4485	0.0843	0.4903	0.0467	0.3840	0.0333
30°	0.4578	−0.0182	0.4640	0.0295	0.4303	0.0518	0.4807	0.0432	0.2905	−0.0078
35°	0.4651	−0.0445	0.4872	0.0253	0.4859	0.0524	0.5025	0.0521	0.3759	0.0391
40°	0.4710	−0.0303	0.5039	0.0501	0.4451	0.0716	0.4926	0.0323	0.3511	0.0380
45°	0.4408	−0.0240	0.4989	0.0285	0.4096	0.0651	0.4810	0.0479	0.3007	0.0072

frames to enhance the contrast. The lighting change is to increase or decrease luminance by 50.

Table 5 shows the simulation results for the combination attacks including RST, compression, salt & pepper noise addition, and lighting change. With the normalized cross-correlation as our detection method, changing the brightness of the target frames has no effect on the simulation results. Therefore, increasing or decreasing the gray scale has the same results for our algorithm. Table 6 shows the simulation results for the combination attacks including RST, compression, salt & pepper noise addition, and histogram equalization. Both Tables 5 and 6 show that there are big and clear separations in the similarity results between the watermarked videos and the unwatermarked videos. This demonstrates that our algorithm is robust against the combination attacks mentioned above.

4.11. Performance comparison

In this section, we provide some discussions on our experimental results by comparing to the two existing video watermarking algorithms [14,15] that are most related to our algorithm. Lu's and our algorithms use the normalized cross-correlation as the detection measure while Wang's algorithm calculates the average error rate to evaluate the robustness against attacks.

Regarding the performance against RST attacks, we have given the experimental results for rotations of up to 45° with cropping. For scaling, we have given the experimental results for conversions among three typical aspect ratios. We also have given the experimental results for translations. All the normalized correlation values are above around 0.6. In comparison, the normalized

Table 6
Combination attacks results II with histogram equalization.

	Mobile		Football		Table tennis		Foreman		Garden	
	Mark	No mark	Mark	No mark	Mark	No Mark	Mark	No Mark	Mark	No Mark
Rotation 0°	0.5931	0.0010	0.4640	0.0291	0.4349	0.0253	0.6010	0.1022	0.3857	0.0581
1°	0.5714	−0.0076	0.4257	−0.0126	0.4256	0.0214	0.5978	0.1055	0.3647	0.0512
2°	0.5754	0.0031	0.4353	0.0039	0.4337	0.0229	0.5952	0.1020	0.3764	0.0625
3°	0.5713	0.0035	0.4528	0.0151	0.4394	0.0241	0.5939	0.1017	0.3864	0.0645
4°	0.5742	0.0050	0.4689	0.0205	0.4333	0.0317	0.5919	0.1043	0.3848	0.0625
5°	0.5653	−0.0008	0.4714	0.0338	0.4433	0.0193	0.5661	0.0899	0.3762	0.0409
10°	0.5513	−0.0175	0.4594	0.0241	0.4634	0.0425	0.5731	0.1018	0.3678	0.0514
15°	0.5333	0.0107	0.4864	0.0197	0.4352	0.0398	0.5496	0.0958	0.3733	0.0306
20°	0.5018	0.0017	0.4850	0.0278	0.4312	0.0298	0.5542	0.1119	0.3678	0.0355
25°	0.4914	−0.0034	0.4848	0.0241	0.4414	0.0568	0.5174	0.0937	0.3499	0.0353
30°	0.4796	−0.0094	0.4499	0.0263	0.4222	0.0195	0.5182	0.0957	0.2619	0.0057
35°	0.4853	−0.0242	0.4642	0.0257	0.4268	0.0168	0.5235	0.0897	0.3440	0.0471
40°	0.4805	−0.0245	0.4486	0.0018	0.4038	0.0370	0.5197	0.0796	0.3196	0.0335
45°	0.4597	−0.0120	0.4559	0.0190	0.3971	0.0402	0.4847	0.0743	0.2791	0.0214

correlation value for Lu's algorithm [14] is 0.23 for a rotation of 15°. The normalized correlation value for rescaling is 0.27. Wang's algorithm uses the bit error rate to measure the performance of watermark detection [15]. The average error rate for Wang's algorithm is 2.13% for slight rotations; and up to 2% for downsizing to 0.7. For our algorithm, we used five different videos to verify the robustness, and we did not find any failures for rotation, scaling and translation attacks.

For the performance against video compression, we have given the experimental results for H.264 compression with different bit rates in Section 4.9. In order to generate PSNRs around 40 dB, the bit rates are varied for different videos, such as, 5.5 Mbps for Mobile, 3.8 Mbps for Football, 3.0 Mbps for Table tennis, 2.4 Mbps for Foreman and 3.7 Mbps for Garden (refer to Tables 3 and 4). The normalized correlation values are between 0.43 and 0.57 with an average of 0.51. For Lu's algorithm [14], the detection result is 0.31 for MPEG compression with a bit rate of 6 Mbps, and 0.24 for a bit rate of 2 Mbps. As for Wang's algorithm [15], an error rate of 0% has been reported for MPEG recompression from a bit rate of 6 to 4 Mbps.

5. Conclusions and future work

This paper presented a robust video watermarking algorithm based on the 1D DFT and Radon transform. This algorithm avoids using 3D transform for video to save computation costs. Embedding fence shaped watermark pattern into frames in the temporal frequency domain can not only re-synchronize after RST attacks, but also spread the watermark information into all frames in the spatial domain to keep the fidelity of the watermarked video. The property that the I frame of one GOP always has bigger change by the watermarking helps this algorithm be robust to H.264 compression. The gradient detection method significantly improves the similarity calculation. One of the most important advantages of this video watermarking algorithm is its simplicity and practicality.

Attackers probably could remove the watermark by deleting the high frequency frames in the temporal frequency domain. However, to remove high frequency frames will significantly reduce the video quality. To improve the security of watermark could be our future work. However, the current algorithm can be practically used for data hiding in many applications such as video broadcasting, indexing, authentication, and enhancement. In addition to improving security, we will also consider attacks such as camera capturing and temporal sampling.

Acknowledgments

The authors would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) for its generous support.

References

[1] D. Zheng, Y. Liu, J. Zhao, A. El Saddik, A survey of RST invariant image watermarking algorithms, ACM Computing Surveys 39 (2) (2007) 1–91.
[2] J. O'Ruanaidh, T. Pun, Rotation, scale, and translation invariant digital image watermarking, Signal Processing 66 (3) (1998) 303–317.
[3] C. Lin, M. Wu, J. Bloom, I. Cox, M. Miller, Y. Lui, Rotation, scale, and translation resilient watermarking for images, IEEE Transactions on Image Processing 10 (5) (2001) 767–782.
[4] C. Deng, X. Gao, D. Tao, X. Li, Geometrically invariant watermarking using affine covariant regions, in: Proceedings of IEEE International Conference on Image Processing, 2008, pp. 413–416.
[5] C. Deng, X. Gao, X. Li, D. Tao, Invariant image watermarking based on local feature regions, in: Proceedings of IEEE International Conference on Cyberworlds, 2008, pp. 6–10.
[6] C. Deng, X. Gao, D. Tao, X. Li, Digital watermarking in image affine co-variant regions, in: Proceedings of IEEE International Conference on Machine Learning and Cybernetics, vol. 4, 2007, pp. 2125–2130.
[7] Y. Liu, D. Zheng, J. Zhao, A rectification scheme for RST invariant image watermarking, IEICE Transactions on Fundamentals, Special Section on Cryptography and Information Security E88-A (1) (2005) 314–318.
[8] G.C. Langelaar, R.L. Lagendijk, Optimal differential energy watermarking of DCT encoded images and video, IEEE Transactions on Image Processing 10 (1) (2001) 148–158.
[9] I. Cox, J. Kilian, T. Leighton, T. Shamoon, Secure spread spectrum watermarking for multimedia, IEEE Transactions on Image Processing 6 (12) (1997) 1673–1687.

[10] F. Hartung, B. Girod, Watermarking of uncompressed and compressed video, Signal Processing 66 (1998) 283–301.
[11] J. Meng, S. Chang, Tools for compressed-domain video indexing and editing, in: Proceedings of the SPIE, Storage and Retrieval for Image and Video Database, vol. 2670, 1996, pp. 180–191.
[12] Z. Zhao, N. Yu, X. Li, A novel video watermarking scheme in compressed domain based on fast motion estimation, in: Proceedings of International Conference on Communication Technology Proceedings, vol. 2, 2003, pp. 1878–1882.
[13] K.-P. Kang, Y.-H. Choi, T.-S. Choi, Real-time video watermarking for MPEG streams, in: Proceedings of the SPIE, vol. 5203, 2003, pp. 691–702.
[14] C.-S. Lu, H.-Y.M. Liao, Video object-based watermarking: a rotation and flipping resilient scheme, in: Proceedings of IEEE International Conference on Image Processing, vol. 2, 2001, pp. 483–486.
[15] Y. Wang, A. Pearmain, Blind MPEG-2 video watermarking robust against geometric attacks: a set of approaches in DCT domain, IEEE Transactions on Image Processing 15 (6) (2006) 1536–1543.
[16] K. Su, D. Kundur, D. Hatzinakos, A content-dependent spatially localized video watermark for resistance to collusion and interpolation attacks, in: Proceedings of International Conference on Image Processing, 2001, pp. 818–821.
[17] B. Mobasseri, M. Sieffert, R. Simard, Content authentication and tamper detection in digital video, in: Proceedings of IEEE International Conference on Image Processing, vol. 1, 2000, pp. 458–461.
[18] B. Mobasseri, Direct sequence watermarking of digital video using m-frames, in: Proceedings of IEEE International Conference on Image Processing, vol. 3, 1998, pp. 399–403.
[19] P. Chan, M.R. Lyu, A DWT-based digital video watermarking scheme with error correcting code, in: Proceedings of Fifth International Conference on Information and Communications Security, 2003, pp. 202–213.
[20] M. Swanson, B. Zhu, A. Tewfik, Data hiding for video-in-video, in: Proceedings of IEEE International Conference on Image processing, vol. 2, 1997, pp. 676–679.
[21] F. Deguillaume, G. Csurka, J.J.K. O'Ruanaidh, T. Pun, Robust 3D DFT video watermarking, in: Proceedings of IS&T/SPIE Electronic Imaging '99, vol. 3657, 1999, pp. 113–124.
[22] Y. Liu, J. Zhao, RST invariant video watermarking based on 1D DFT and Radon transform, in: Proceedings of the 5th IET Visual Information Engineering 2008 Conference (VIE'08), 2008, pp. 443–448.
[23] H. Kim, Y. Baek, H. Lee, Y. Suh, Robust image watermark using Radon transform and bispectrum invariants, Lecture Notes in Computer Science, vol. 2578, 2003, pp. 145–159.
[24] J. Vidal, M. Madueño, E. Sayrol, Color image watermarking using channel-state knowledge, in: Proceedings of SPIE, vol. 4675, 2002, pp. 214–221.
[25] D. Zheng, J. Zhao, A. El Saddik, RST invariant digital image watermarking based on log-polar mapping and phase correlation, IEEE Transactions on Circuits and Systems for Video Technology 13 (8) (2003) 753–765.
[26] Y. Liu, D. Zheng, J. Zhao, An image rectification scheme and its applications in RST invariant digital image watermarking, Multimedia Tools and Applications 34 (1) (2007) 57–84.