



A geometrically robust multi-bit video watermarking algorithm based on 2-D DFT

Xue-Cheng Sun¹ · Zhe-Ming Lu^{1,2,3} · Zhe Wang¹ · Yong-Liang Liu²

Received: 19 June 2020 / Revised: 28 September 2020 / Accepted: 22 December 2020 /
Published online: 15 January 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC part of Springer Nature 2021

Abstract

The development of the Internet, together with the progress of multimedia processing techniques, has led to the problems of data piracy, data tampering and illegal dissemination. Digital watermarking is an effective approach to data authentication and copyright protection. This paper proposes a geometrically robust multi-bit video watermarking algorithm based on 2-D DFT (two-dimensional discrete Fourier transform). While most of the existing video watermarking schemes require synchronization to extract the watermark from rotated or scaled videos, which is time-consuming and affects the accuracy, the proposed method can do direct extraction without performing synchronization for videos attacked by rotation, scaling or cropping. For embedding the watermark, circular templates in DFT domain are transformed into spatial masks and added to the video frames in spatial domain. A perceptual model based on local contrast is applied to keep the fidelity of the watermarked video. We also propose an accurate and efficient extraction method which is based on the cross-correlation between the Wiener-filtered DFT magnitude and the stretched template sequence in polar coordinates. Experimental results show that the proposed algorithm is robust against various kinds of attacks, such as compression, filtering, rotation, scaling, cropping, frame averaging and frame rate changing.

Keywords Video watermarking · Robust watermarking · 2-D DFT · Geometric attacks

✉ Zhe-Ming Lu
zheminglu@zju.edu.cn

¹ School of Aeronautics and Astronautics, Zhejiang University, Hangzhou 310027, People's Republic of China

² Alibaba Group, Hangzhou 311121, People's Republic of China

³ State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, People's Republic of China

1 Introduction

Digital multimedia data are spreading widely and rapidly via the Internet. Meanwhile, problems of data piracy, data tampering and illegal dissemination are becoming increasingly serious, and hence copyright protection has received much concern in the digital world. Digital watermarking is a powerful tool for protection of copyrights on multimedia product. It is a technique of hiding a piece of data called watermark into digital multimedia content called host data [4]. The host data can be of any type, including text, image, audio, video and software. In terms of robustness with respect to transformations of host data, watermarking techniques can be classified into two types: robust watermarking and fragile watermarking. Robust watermarks are resistant to malicious attacks and often used in copyright protection to declare rightful ownership, while fragile watermarks are very sensitive and commonly used for integrity proof. In the present study, we only focus on the robust watermarking technique.

Most of the research has dealt with image watermarking. Image watermarking algorithms can be classified as spatial domain algorithms and transform domain algorithms. Spatial domain algorithms [5, 11, 15] embeds the watermark by modifying the gray levels of some subsets of image pixels. These methods have the advantages of simple operation and high speed. However, transform domain algorithms have attracted more research interest because of their higher robustness and stability [20]. Such algorithms insert the watermark into transformed coefficients of images. The most commonly used transform methods are the discrete cosine transform (DCT), discrete wavelet transform (DWT) and discrete Fourier transform (DFT). DCT based algorithms [19, 22] and DWT based algorithms [1, 23] follow the same guidelines, and they are robust against simple image processing attacks such as compression, low pass filtering and noise addition. Compared with DCT and DWT, DFT has the advantage of being invariant to rotation, scaling and translation (RST), which are considered as the most challenging attacks for image watermarking. Some researches on DFT based algorithms are briefly described in the following.

Solachidis and Pitas [21] proposed a DFT based approach for RST invariant watermarking of digital images. A circularly symmetric watermark is embedded in the magnitude of DFT domain. Since the watermark is circular in shape and symmetric with respect to the image center, it is robust against geometric rotation attacks.

Pramila, Keskinarkaus and Seppänen [18] proposed a watermarking algorithm that takes advantage of multiple watermarking. A circular template consisting of 0's and 1's arranged around the origin symmetrically is embedded in the magnitude of DFT domain. Once the image undergoes geometric transformations this template is searched to resynchronize the image. The robust message watermark embedded in wavelet domain is extracted after the inversion of geometric transformations.

Liao, Li and Yin [10] proposed a method of separable data hiding in encrypted images by using compressive sensing and discrete Fourier transform. This method takes full advantage of both real and imaginary DFT coefficients for ensuring great recovery of the original image and providing flexible payload. The data is embedded into the high-frequency coefficients by using compressive sensing. Compared with the previous work, the quality of decrypted image is greatly improved when concealing the same embedding capacity.

Urvoy, Goudia and Autrusseau [24] proposed a novel DFT based watermarking algorithm featuring perceptually-optimal visibility versus robustness. The watermark is a noise-like square patch of coefficients embedded by substitution in Fourier domain. A perceptual model of human visual system (HVS) based on the contrast sensitivity function and a local contrast

pooling is used to determine the optimal watermark strength. This method is highly robust against various kinds of attacks, especially geometric distortions.

Compared with image watermarking, video watermarking is a much more challenging task. There is a large amount of redundant data between video frames, thus the compression ratio for videos is usually higher than that of images, which requires the watermarking algorithm to be much more robust. Another reason for the difficulty of video watermarking is that besides general image processing, a good video watermarking algorithm must be robust against many other attacks, including frame rate changing, frame swapping, geometric distortions and so on. Lastly, a low time complexity is also necessary. A one-minute video clip can contain more than one thousand frames, suggesting it is hard to implement an ordinary image watermarking algorithm for each video frame without optimization in time.

Video watermarking techniques can be classified into two types: compressed domain watermarking and uncompressed domain watermarking. Compressed domain watermarking algorithms have very high speeds because they only partly decode the video streams, but such techniques are usually limited to a specific compression standard and not robust enough to resist practical video processing attacks [6, 14, 16]. On the other hand, although watermarking algorithms in uncompressed domain are time-consuming, they are much more robust against various kinds of distortions. For example, Liu and Zhao [13] proposed a zero-bit video watermarking algorithm robust to RST attacks, by using 1-D DFT in temporal direction and Radon transform in temporal frequency domain. Faragallah [7] proposed an efficient, robust and imperceptible semi-blind video watermarking scheme based on DWT and singular value decomposition. Chen and Zhao [2] proposed a blind video watermarking scheme, in which the watermark is embedded to the principal components of the largest contourlet coefficients of the last directional subband. However, the above algorithms are far from perfect, because they require synchronization to extract the watermark from rotated or scaled videos, which may be an exhaustive search and is too time-consuming to be used in practice.

Based on the above, we propose a geometrically robust video watermarking algorithm based on 2-D DFT to efficiently embed and extract multi-bit watermarks. This algorithm is processed in uncompressed domain, so it is suitable for most video compression standards. The watermark is a sequence of circular templates consisting of 0's and 1's symmetrically arranged around the center of the magnitude of DFT domain. In order to reduce the time complexity, the algorithm transforms the circular templates in DFT domain to spatial masks independent of video frames and embeds the watermark in spatial domain. Since embedding is processed in spatial domain, a perceptual model based on local contrast of video frames can be easily applied to improve the invisibility of the watermark. To embed multi-bit watermarks, the proposed scheme segments the video frames into groups of pictures (GOPs). The frames from one GOP contain the same template and each of the GOPs holds one watermark bit. For watermark extraction, the proposed algorithm never needs to perform synchronization. It is only necessary to perform one DFT for each GOP and then compute the cross-correlation between the Wiener-filtered DFT magnitude and the stretched template sequence in polar coordinates, which is not only efficient but very accurate for template detection.

The main contributions of this paper can be summarized as follows: (1) A sequence of DFT templates is used as the watermark to handle geometric attacks; (2) The DFT templates are transformed to spatial masks to speed up embedding and facilitate the implementation of the local contrast-based perceptual model; (3) The extraction procedures including DFT, Wiener filtering and cross-correlation in polar coordinates are fully optimized to improve the computational efficiency and the accuracy of extraction.

The rest of this paper is organized as follows. Section 2 gives a brief overview of DFT and introduces the preliminaries of DFT template. Section 3 details the embedding and extraction steps of the proposed method. Section 4 presents the experimental results. Section 5 gives the conclusions of this paper and discusses about the future directions.

2 Preliminaries

2.1 2-D DFT and its properties

The discrete Fourier transform (DFT) is a frequency domain representation of finite-extent sequences [12]. Let the image be a real-valued function $f(x, y)$ defined on an integer-valued Cartesian grid $0 \leq x < M$, $0 \leq y < N$. The 2-D DFT of f is defined as:

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi xu/M - j2\pi yv/N} \quad (1)$$

where $0 \leq u < M$, $0 \leq v < N$. The inverse transform (IDFT) is:

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi ux/M + j2\pi vy/N} \quad (2)$$

The first important property of DFT is that it is linear:

$$af(x, y) + bg(x, y) \leftrightarrow aF(u, v) + bG(u, v) \quad (3)$$

This is convenient, since when the watermark is added to the host image in DFT domain, adding the IDFT of the watermark in spatial domain can achieve the same effect.

The rotation property of 2-D DFT is also useful. This means that rotating spatial domain contents by θ rotates frequency domain contents by θ . This can be formally described by the following relationship:

$$\begin{aligned} f(x, y) &\leftrightarrow F(u, v) \Leftrightarrow f(x', y') \leftrightarrow F(u', v') \\ \begin{pmatrix} x' \\ y' \end{pmatrix} &= \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}, \begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} \end{aligned} \quad (4)$$

The scaling property of DFT also plays an important role in image watermarking. This means that shrinking in one domain causes expansion in the other. The equation describing this is:

$$f(ax, by) \leftrightarrow \frac{1}{ab} F\left(\frac{u}{a}, \frac{v}{b}\right) \quad (5)$$

One last property of DFT is its translation invariance. In fact, spatial translations or shifts only affect the phase component of DFT, thus the magnitude of DFT is translation invariant. As can be seen in the following subsection, this property can directly lead to the watermark's robustness against cropping. The equation governing this property is:

$$f(x-x_0, y-y_0) \leftrightarrow F(u, v) e^{-j2\pi ux_0/M - j2\pi yv_0/N} \quad (6)$$

2.2 The circular template in DFT domain

The DFT template forms the basis of the proposed algorithm. As in [18], it is a circular sequence consisting of 0's and 1's symmetrically arranged around the center of the magnitude of DFT domain. The basic principles are illustrated in the following simple example.

Suppose the 0–1 template sequence is an array of length l that alternates between 0 and 1. After transforming the image using DFT, we first shift the magnitude component so that the zero frequency is at the center. To balance the robustness and invisibility of the template, a ring at the middle frequencies is chosen as the embedding region. Then we pick l equally spaced points corresponding to the 0–1 array from the upper half of the ring. For each 1 of the array, a constant C is added to the magnitude of the DFT coefficient at its corresponding point on the ring. The watermarked DFT magnitude must be symmetric with respect to the center to ensure the IDFT is real [21], thus we also add the constant to the magnitudes at the symmetric points on the lower half of the ring. As a result, a sparse circle that contains l equally spaced points is added to the magnitude of DFT. The watermarked magnitude of the 512×512 Lena image is shown in Fig. 1.

As discussed in the previous subsection, the DFT template is robust against rotation and scaling. Rotation in spatial domain causes rotation in Fourier domain by the same angle, and shrinking in spatial domain causes expansion in Fourier domain. Thus for a rotated or scaled image, the DFT template is still a circular sequence symmetrically arranged around the center, which means an extraction method that can detect the rotated or scaled circle is all we need. Another important property of the DFT template is that it is invariant to cropping. This means for a cropped image, the template is still the same sequence at the same frequencies. An intuitive explanation of this is that cropping is basically a kind of shifting, so the magnitude of

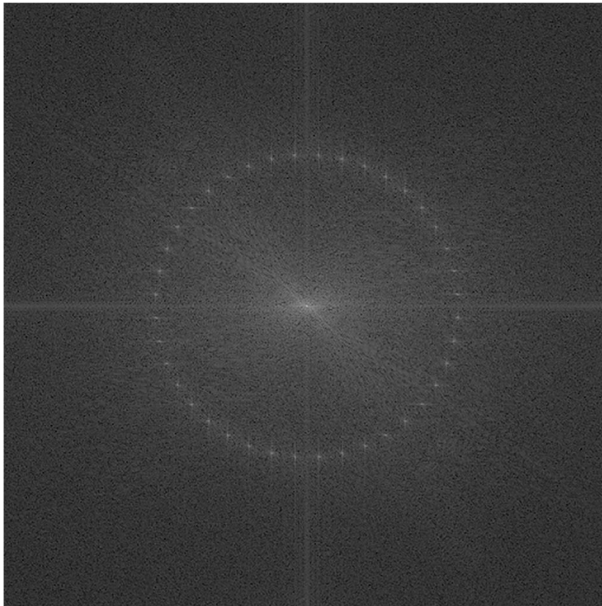


Fig. 1 Watermarked DFT magnitude of the 512×512 Lena image. The template sequence is an array of length 40 that alternates between 0 and 1

DFT is not changed; and because cropping does not change the sampling frequency, the sequence is still at the same frequencies. As an example, the templates of the 512×512 Lena images attacked by rotation, scaling and cropping respectively are shown in Fig. 2.

The above procedures for embedding the DFT template work well for images, but are not suitable for videos, because it is too time-consuming to calculate the DFT for a large number of video frames. Fortunately, we can implement the embedding in spatial domain by using the linearity of DFT. More specifically, we first embed the template in a single-valued gray scale image to obtain the difference between the watermarked and the original image, then for each of the host images, we can simply add the difference image to it in spatial domain. We call the difference image spatial mask in this paper. The spatial mask is basically an overlapping of sinusoidal waves, and in fact, a large enough spatial mask can be used for all images with a smaller size because of the cropping invariance property of the DFT template. The detailed procedures of generating the spatial mask are as follows.

- Step: 1. Generate an $L \times L$ ($L = 5000$ is enough in most cases of video watermarking) gray level image I , in which the gray values of all pixels are 128.
- Step: 2. Calculate the DFT of I . The magnitudes of all coefficients other than the direct current (DC) component are zero. Then for a given 0–1 sequence, embed the corresponding circularly symmetric template in Fourier domain through the method presented above, and obtain the watermarked I by calculating the IDFT using the watermarked magnitude.

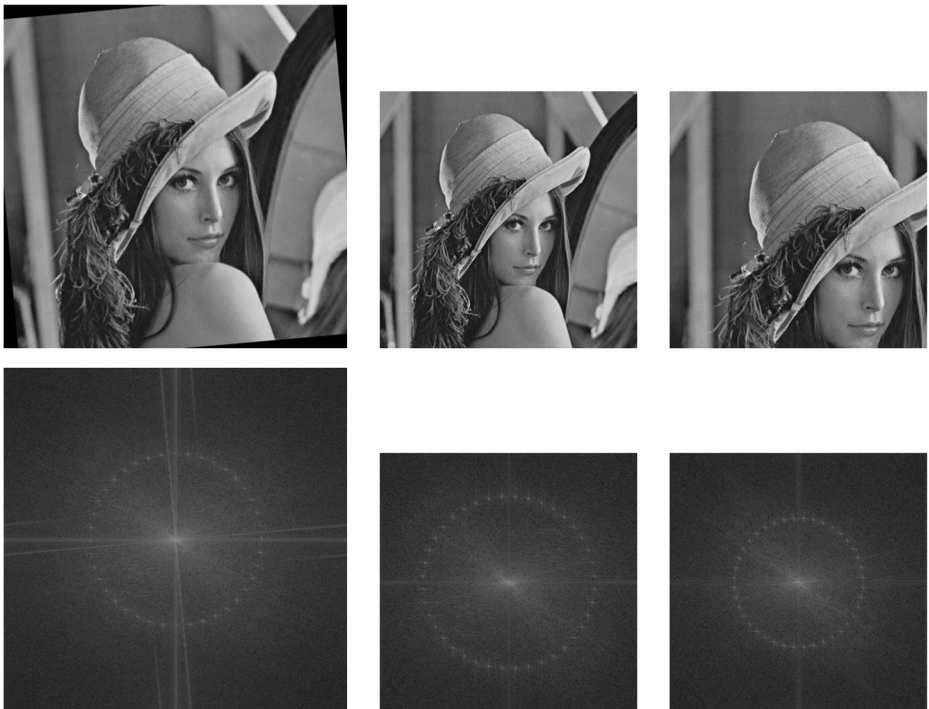


Fig. 2 Geometrically attacked images (top) and their DFT templates (bottom). The attacks from left to right are rotation by 5° , downscaling from 512×512 to 384×384 and cropping that only keeps the 384×384 upper-left corner. All images in this figure are scaled to represent their true relative sizes

- Step: 3. Obtain the $L \times L$ spatial mask by subtracting the original I from the watermarked I .
- Step: 4. To embed the template in an $M \times N$ ($M, N \leq L$) image J , just add the central $M \times N$ region of the spatial mask to J in spatial domain. For color images, the template is embedded in the Y component of the YUV decomposition.

The main advantage of using the spatial mask is that we can avoid the heavy computation of DFT. Also, once we have found a suitable strength for the sparse circle embedded in the DFT magnitude, we never need to change it. To keep the fidelity of the watermarked image, it is only necessary to adjust the strength of the mask in spatial domain, which can be achieved by the perceptual model proposed in the next section.

3 Proposed method

The proposed video watermarking algorithm consists of two parts: watermark embedding and watermark extraction. Figure 3 shows the block diagram of the algorithm. In the embedding process, the original video is first divided into groups of pictures (GOPs), each of which hosts exactly one watermark bit. For each GOP, we adjust the strength of the corresponding spatial mask by using a perceptual model and then add it to the frames of the GOP successively. To extract the watermark, we first calculate the DFT of the average image of each GOP. Then the DFT magnitudes are processed by the Wiener filter, and the filtering results are correlated in polar coordinates with the stretched template sequences to determine the watermark bits. In the next two subsections, the principle and implementation of the algorithm will be explained in detail.

3.1 Watermark embedding

In this subsection, we introduce the embedding process, in which the spatial masks are added to successive video frames after being adjusted by the perceptual model based on local contrast of the frames.

In image watermarking, the DFT template is usually used as a synchronization pattern to invert geometric transformations, and a message watermark needs to be embedded in another

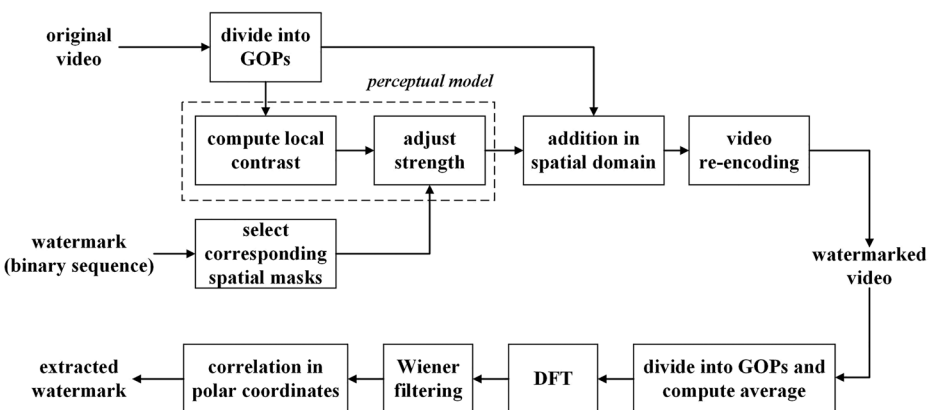


Fig. 3 Block diagram of the proposed algorithm

domain [18]. The drawbacks of such scheme are that images containing multiple watermarks often have low fidelities and the extraction may fail due to the limited resolution of the DFT template and the inaccuracy of the synchronization process. In video watermarking, we can avoid the use of message watermark by embedding different templates into different groups of frames. The watermark can always be encoded into a bit sequence, thus we only need two kinds of DFT templates: the zero-template for watermark bit zero and the one-template for watermark bit one. The only difference between the two templates is that we use different 0–1 sequences to generate the sparse circles, which can be called zero-template sequence and one-template sequence respectively. To embed a multi-bit watermark into a given video, we first segment the frames into groups of pictures (GOPs), each of which can host one watermark bit. Then for every frame within a GOP, we embed the zero-template into it (by adding the spatial mask) if the corresponding watermark bit is zero, and embed the one-template otherwise. It is clear that the capacity of the watermark is equal to the total number of GOPs.

The choice of the number of frames in each GOP is the first key point during embedding. In an extreme case, each GOP contains only one frame. This can maximize the watermarking capacity, but the watermark is sensitive to attacks like frame swapping and frame averaging. Also we cannot use the same number all the time, because such scheme is not able to resist the frame rate changing attack. A better solution is to choose the number according to the frame rate of the host video. Suppose that the frame rate is r frames per second (fps), then we can set the number of frames in each GOP to r , i.e., there is one GOP per second and one second of time can host one watermark bit. For most real world videos, the frame rate is above 20 fps, thus the GOPs are long enough to resist frame swapping and frame averaging. Later in the extraction process, the watermarked video will also be segmented based on its frame rate, which guarantees that the temporal synchronization of the watermark will not be destroyed by frame rate changing.

In the design of embedding scheme, we also have to consider the effect of inter-frame coding used in video compression. Inter-frame coding is a kind of predictive coding method which encodes the differences between frames rather than each full frame. For a given frame, if it falls into a different GOP from its reference frames, then the watermark embedded in the reference frames can probably be detected in the given frame as well. The reason is that the details of the differences are partly destroyed by compression, which makes it very difficult to completely erase the watermark of the reference frames when we decode the given frame. According to our experiments, this phenomenon is particularly evident when there is an intra reference frame in which the complete image is coded and when the video is downscaled or compressed with a high ratio. An example of this is shown in Fig. 4. Only the template in Fig. 4(a) is embedded into the given frame, but after the video is downscaled, the two templates are mixed together. In the most serious cases, this can be observed in more than half of the frames in one GOP, which will certainly affect the extraction accuracy. To solve this problem, we can simply insert an interval with a length of $0.4r$ between every pair of GOPs. The mixing phenomena will occur mostly in intervals, but they will be skipped in the extraction process. It is important to note that $0.4r$ is not always an integer, thus we need to make a correction every 5 intervals ($2r$ frames) to guarantee the average duration of the intervals is strictly 0.4 s, otherwise there will be a risk of large accumulative synchronization errors during extraction.

Setting appropriate 0–1 template sequences is another key point. It should be easy for the extraction process to distinguish between the zero-template sequence s_0 and the one-template sequence s_1 . Suppose that s_0 is a fixed sequence of length l . One strategy is to set s_1 as the complement to s_0 to minimize the normalized correlation (NC) value between them, which is defined as:

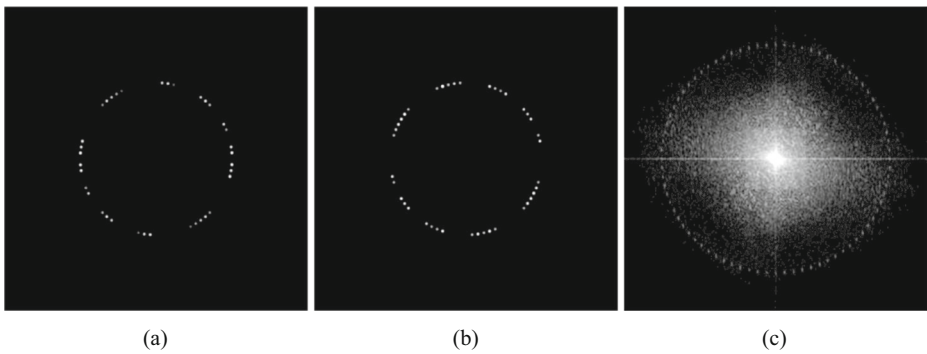


Fig. 4 An example of the mixing phenomenon. **a** The circular template that is embedded into the given frame. **b** The circular template that is embedded into the frames of the previous GOP. In this case, there is a key frame at the end of the previous GOP. **c** The DFT magnitude of the given frame after the video is downscaled from 1920×1080 to 1280×720

$$NC(s_0, s_1) = \frac{\sum_{i=0}^{l-1} s_0(i) \cdot s_1(i)}{\sqrt{\sum_{i=0}^{l-1} s_0^2(i)} \sqrt{\sum_{i=0}^{l-1} s_1^2(i)}} \tag{7}$$

An example of this is shown in Fig. 4(a) and (b). The two sequences are

$$\begin{aligned} s_0 &= 0101100111000111100001111100000111111000 \\ s_1 &= 1010011000111000011110000011111000000111 \end{aligned} \tag{8}$$

Clearly the NC between s_0 and s_1 is zero. However, the sequences can be circularly shifted by rotation of video frames, resulting in increase of the NC value. For example, if s_1 is shifted to the left by five bits:

$$s_1' = 1100011100001111000001111100000011110100 \tag{9}$$

then the NC will become 0.7, which means the extractor may have difficulty in distinguishing between the two sequences. A simple way to solve this problem is to set the two sequences as follows:

$$s_0(i) = \begin{cases} 0 & \text{when } i \bmod 2 = 0 \\ 1 & \text{when } i \bmod 2 = 1 \end{cases} \quad \text{for } i = 0, 1, \dots, l-1 \tag{10}$$

$$s_1(i) = \begin{cases} 0 & \text{when } i \bmod 4 = 0 \text{ or } i \bmod 4 = 1 \\ 1 & \text{when } i \bmod 4 = 2 \text{ or } i \bmod 4 = 3 \end{cases} \quad \text{for } i = 0, 1, \dots, l-1 \tag{11}$$

Obviously, the NC value is always 0.5 no matter how the two sequences are circularly shifted. The circles corresponding to these two sequences are shown in Fig. 5.

The last step of embedding is to implement a perceptual model to improve the fidelity of the watermarked video. Since the watermarks are essentially additive noise superimposed on the frames, the noise masking methods proposed in [1, 3] can be easily applied. However, these methods depend on a number of parameters, which require lots of fine-tuning and often lead to

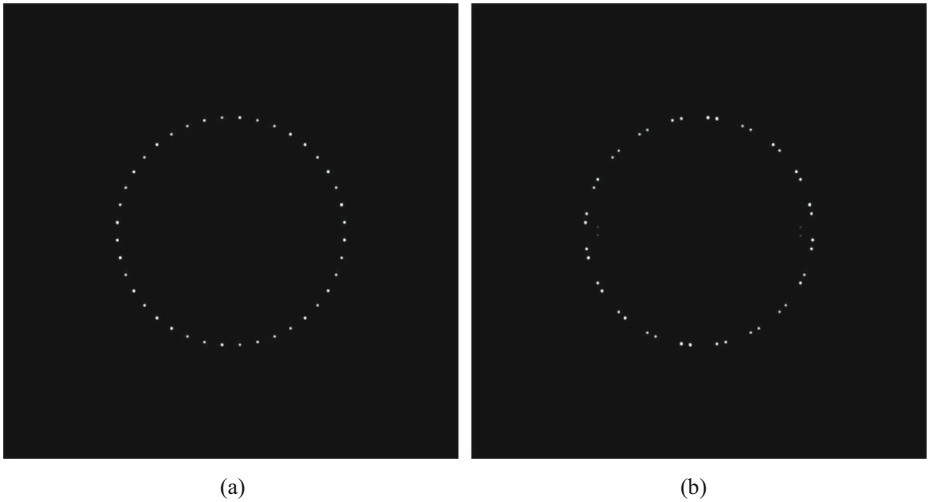


Fig. 5 How s_0 and s_1 are arranged in the DFT magnitude of an image. **a** s_0 . **b** s_1

unstable results. Here we propose a simple but effective perceptual model based on the local contrast of images. The contrast c at position (x, y) is calculated as follows:

$$c(x, y) = \sum_{i=0}^1 \sum_{j=0}^1 p(x + i, y + j) \cdot D(i, j) \tag{12}$$

where $p(x, y)$ denotes the pixel value at (x, y) and D denotes the operator for computing the contrast:

$$D = \begin{bmatrix} -1 & -1 \\ -1 & 3 \end{bmatrix} \tag{13}$$

The local contrast lc around position (x, y) is the average of the absolute contrast in the 5×5 area centered at (x, y) plus a constant a :

$$lc(x, y) = a + \frac{1}{25} \sum_{i=-2}^2 \sum_{j=-2}^2 |c(x + i, y + j)| \tag{14}$$

Let $m(x, y)$ denote the pixel value of the spatial mask at position (x, y) . The intensity of the mask is adjusted according to the following rule:

$$m'(x, y) = \begin{cases} b \cdot T \cdot m(x, y) & \text{if } lc(x, y) > T \\ b \cdot lc(x, y) \cdot m(x, y) & \text{if } lc(x, y) \leq T \end{cases} \tag{15}$$

where T is a threshold and b is the parameter that controls the overall intensity. The principle behind this is that human eyes are more sensitive to noise in smooth regions and less sensitive to noise in regions with high contrast. And since the level of local contrast may differ greatly for different images, we need a threshold that limits the maximum intensity of the mask to make the model more stable. The value of the overall intensity should be set according to the desired fidelity of the watermarked video; recommendation for the other parameters is $a = 10$ and $T = 25$.

The embedding procedures can be summarized as follows.

- Step: 1. Construct the zero-template sequence and the one-template sequence according to Eqs. 10 and 11. Then generate the two spatial masks using the approach described in the previous section. The results can be stored, so this step does not need to be performed again for a different input video.
- Step: 2. Divide the input video into GOPs with the number of frames in each GOP equal to the frame rate r . There should be an interval with a length of $0.4r$ between every pair of GOPs.
- Step: 3. Embed one bit of watermark into each of the GOPs by adding the corresponding spatial mask to all frames of each GOP. The strength of the spatial masks should be adjusted by the perceptual model before every time of adding.
- Step: 4. Re-encode the frames to obtain the watermarked video.

3.2 Watermark extraction

In this subsection we introduce the extraction process. The goal of extraction is to identify which DFT template is embedded into each of the GOPs. The video is first divided into GOPs in the same way as the embedding process. Unlike embedding, the detection of DFT template has to be performed in frequency domain, but it takes too long to implement a DFT for every frame. Fortunately, we can make use of the linearity of DFT to speed up this process. Assume that the same template is embedded into all frames of a GOP. If we compute the average of the frames and calculate the DFT magnitude of it, the result will be the average of the DFT magnitudes of the frames plus the embedded template. Then to extract the bit of watermark embedded in the GOP, we just need to perform one circle detection algorithm. According to experiments, this trick can increase the speed of extraction by more than four times without affecting the accuracy.

The template is essentially additive noise superimposed on the magnitude image, thus the first step of circle detection is to filter the magnitude with a noise extraction filter. As in [18], we use the adaptive Wiener filter which is a minimum mean square error estimator based on statistics estimated from a local neighborhood of each pixel [9]. The original Wiener filter is a low-pass filter that reduces additive noise, thus we need to subtract the filtering result from the magnitude image. To suppress the pixels that do not contain the watermark information, a threshold operation is then performed on the result of subtraction. So far, the implementation of Wiener filtering is the same as in [18], but it needs several additional steps to handle the stronger attacks on videos. First, values of the pixels on the horizontal and vertical lines at the middle of the magnitude image should be replaced with zeros. The reason for the appearance of the two lines is that DFT has an implicit periodicity in both spatial and frequency domains. For most of the natural images, there are strong horizontal and vertical discontinuities at the periodical boundaries, thus the intensities of the pixels on the two lines are often stronger than those of the pixels containing the watermark and need to be suppressed before the threshold operation. Similarly, pixels at the low frequencies also have strong intensities and often affect the threshold result. To handle this, we define a parameter R and values of all the pixels in the circular area of radius R centered at the origin are replaced with zeros. For an $2L \times 2L$ image, we recommend to set R to $0.35L$ if the video is not subject to enlarging attacks; otherwise we should use a smaller R to ensure that watermarks in enlarged videos will not be erased by this operation. The effects of these two additional operations are shown in Fig. 6.

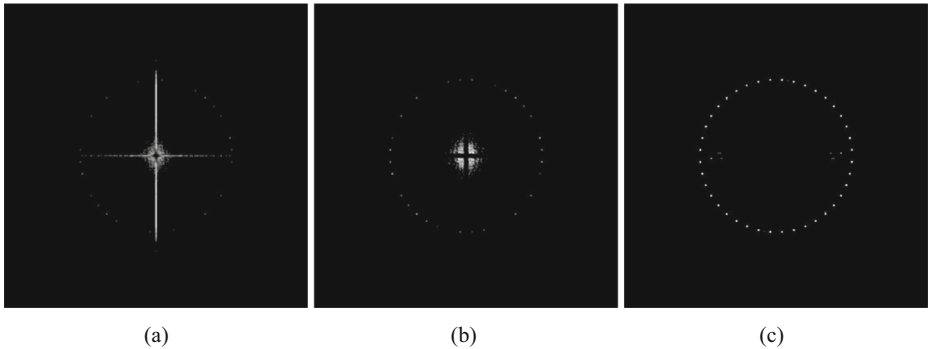


Fig. 6 The effects of the two additional operations in Wiener filtering. **a** The result of the method in [18]. Only a portion of the pixels containing the watermark are extracted. **b** The result obtained by suppressing the pixels on the two lines before the threshold operation. More of the pixels containing the watermark are extracted, but the pixels at the low frequencies are still disturbing the result. **c** The result obtained by also suppressing the pixels at the low frequencies. All of the pixels that contain the watermark are extracted by the threshold operation

The circle is detected by calculating the cross-correlations between the template sequences and the one-dimensional sequences each of which represents a ring with a thickness of 1 pixel in the Wiener-filtered magnitude image. The image is first remapped to polar coordinates so that a ring centered at the origin becomes a row of the resulting matrix. Then we calculate the cross-correlations between a template sequence and each row of the remapped image:

$$\text{corr}_{XY}(n) = \sum_{m=0}^{M-1} X(m)Y(m+n) \quad (16)$$

where X represents a row of the remapped image, M represents the length of X , Y represents the template sequence stretched to the length of $0.5M$ and zero padding is used for the values with out-of-range indices. In fact, we can calculate all of the correlations by computing a two-dimensional convolution between the remapped image and the stretched sequence. As in [18], the scaling factor and rotation angle can be computed by locating the maximum of the result of convolution. But to extract the watermark, we only need to convolve the remapped image with the zero-template sequence and the one-template sequence, find the maximums of both results and choose the bit corresponding to the larger of the two maximums. The advantage of this method over those of [17, 18] is mainly that it does not need to detect local peaks of the magnitude image, which requires an additional threshold parameter and thus does not always produce accurate results.

The extraction procedures can be summarized as follows.

- Step: 1. Divide the input video into GOPs in the same way as the embedding process.
- Step: 2. For each GOP, calculate the average of all its frames and obtain the Fourier magnitude by applying DFT to the average image. The average image needs to be padded to a square before calculating the DFT to prevent the template from being stretched to an eclipse.
- Step: 3. Filter the magnitude image with Wiener filter and subtract the output from it. Then suppress the “noise” pixels in the subtraction result by setting values of the pixels on the two middle lines and the pixels at the low frequencies to zeros. Now if the value of a pixel does not exceed a predefined threshold, it is replaced with a zero.

- Step: 4. Remap the threshold result to polar coordinates. To extract the watermark bit, first convolve the remapped image with the stretched zero-template sequence and one-template sequence, then find the maximums of both results and choose the bit corresponding to the larger of the two maximums. If the maximums are lower than a predefined threshold, the bit of watermark does not exist or the extraction fails.
- Step: 5. Obtain the complete watermark by repeating steps 2 to 4 for each of the GOPs.

4 Experiments

In this section we illustrate the performance of the proposed algorithm.

4.1 Experimental setup

The algorithm is implemented on a 3.6 GHz Intel i7 processor with 16 GB RAM, and is realized by C++ language with the OpenCV library [8]. We first test four CIF videos, and the watermark is a 16-bit binary sequence. The host videos are originally uncompressed YUV sequences, and our watermarking system encodes them into compressed videos using an H.264 encoder. The average bit rate of the encoded videos is set to 512 kbps (kilobits per second). The characteristics of these videos are shown in Table 1. In order to evaluate the algorithm's performance on modern high-definition videos, we also test four 1080p video clips whose sources are freely redistributable movies [26]. We choose the official H.264 versions of the movies and only use the first 90 s of each movie in our experiments. The watermark embedded into these videos is a 64-bit binary sequence. Since the proposed algorithm needs 1.4 s of time to accommodate a bit of watermark, a length of 90 s is enough for a 64-bit sequence. The output of the watermarking system is also an H.264 video, and the average bit rate of the watermarked video is set to the same as the input. The characteristics of the 1080p videos are shown in Table 2. The snapshots of all the eight test videos are shown in Fig. 7.

4.2 Execution efficiency

In this subsection, simulation experiments are carried out using different strategies to show the efficiency of the proposed algorithm. The average embedding/extraction time per frame is calculated for each strategy. Two embedding strategies are tested. The first is to embed the circular template in Fourier domain and the second is to add the noise mask in spatial domain, and we denote them as Embed_1 and Embed_2, respectively. Also, we test three extraction strategies that perform different times of DFT and/or Wiener filtering. In the first strategy denoted as Extraction_1, a DFT followed by a Wiener filter is applied to every frame of each GOP, and the

Table 1 Characteristics of the CIF videos

Video name	Bridge close	Bridge far	Mad900	Students
Size	352 × 288	352 × 288	352 × 288	352 × 288
Total number of frames	2001	2101	900	1007
Frame rate (fps)	29.97	29.97	29.97	29.97
Color space	YUV420p	YUV420p	YUV420p	YUV420p
Bit rate after being watermarked (kbps)	521	527	530	519

Table 2 Characteristics of the 1080p videos

Video name	Big buck bunny	Elephants dream	Sintel trailer	Sita sings the blues
Size	1920×1080	1920×1080	1920×818	1920×1080
Frame rate (fps)	30	24	24	23.976
Color space	YUV420p	YUV420p	YUV420p	YUV420p
Bit rate before being watermarked (kbps)	2765	9165	11,402	7066
Bit rate after being watermarked (kbps)	2686	8884	10,774	7040

rest of the algorithm is performed on the average of the filtering results. In the second strategy denoted as Extraction_2, a DFT is performed on every frame of each GOP, and a Wiener filter is applied to the average Fourier magnitude. Finally, in the last strategy denoted as Extraction_3, both of DFT and Wiener filtering are performed only once on the average image of each GOP. The results for CIF and 1080p videos are shown in Tables 3 and 4, respectively. As can be seen in the tables, reducing the times of performing DFT can significantly improve the time efficiency. In the subsequent experiments, we will only use the strategies of Embed_2 and Extraction_3, which are the most efficient and consistent with the description in the previous section.

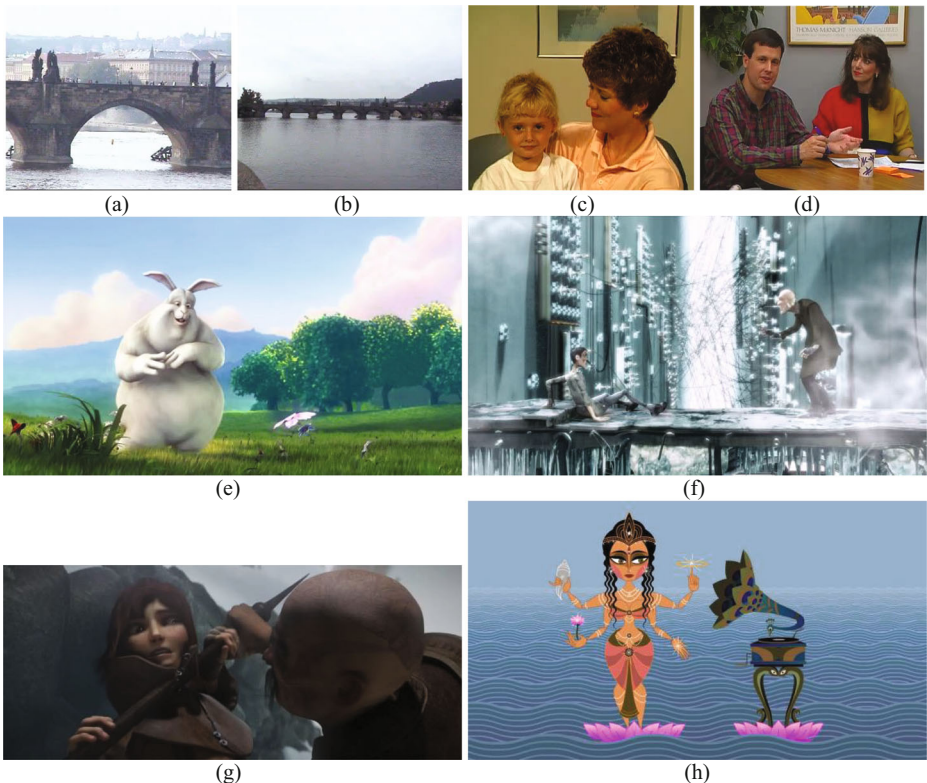


Fig. 7 Snapshots of the eight test videos. **a** Bridge close. **b** Bridge far. **c** Mad900. **d** Students. **e** Big buck bunny. **f** Elephants dream. **g** Sintel trailer. **h** Sita sings the blues

Table 3 Time efficiency for the CIF videos (milliseconds per frame)

	Bridge close	Bridge far	Mad900	Students
Embed_1	4.31	4.14	4.26	4.25
Embed_2	1.89	1.75	1.77	1.76
Extraction_1	8.27	8.09	8.03	8.28
Extraction_2	3.21	3.29	3.21	3.19
Extraction_3	1.26	1.21	1.18	1.19

4.3 Quality metrics

The peak signal-to-noise ratio (PSNR) is used to measure the fidelity of the watermarked videos. Given the original image I and the watermarked image K , the PSNR (in dB) is defined as:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{255^2}{\frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (I(i, j) - K(i, j))^2} \right) \quad (17)$$

where m and n denote the height and width of I , respectively. We calculate the PSNR for the Y channel of YUV color space, and report the average PSNR of all the watermarked frames. Typically, a watermarked image with a PSNR of about 40 dB looks very close to the original one, so in our experiments the PSNR values of the watermarked videos are controlled around 40 dB by adjusting the overall intensity parameter of the perceptual model. The PSNR values of the watermarked videos are shown in Table 5.

Moreover, the structure similarity index measure (SSIM) proposed in [25] is also used as a video fidelity measurement. It is a perception-based index that considers image degradation as perceived change in structural information. The SSIM between images x and y is defined as:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (18)$$

where μ_x is the average of x , μ_y is the average of y , σ_x^2 is the variance of x , σ_y^2 is the variance of y , σ_{xy} is the covariance of x and y , $c_1 = 6.5025$ and $c_2 = 58.5225$ are the two variables used to stabilize the division with weak denominator. The SSIM value ranges between 0 to 1 and only equals 1 if the two images are identical. As in the case of PSNR, we calculate the SSIM for the Y channel and report the average of all the watermarked frames in Table 5.

Table 4 Time efficiency for the 1080p videos (milliseconds per frame)

	Big buck bunny	Elephants dream	Sintel trailer	Sita sings the blues
Embed_1	168.08	172.95	172.02	171.00
Embed_2	44.58	47.32	36.65	47.46
Extraction_1	312.67	317.04	313.59	315.45
Extraction_2	119.59	127.80	125.63	130.53
Extraction_3	35.71	43.44	42.48	43.10

Table 5 Fidelity of the watermarked videos

	Bridge close	Bridge far	Mad900	Students	Big buck bunny	Elephants dream	Sintel trailer	Sita sings the blues
PSNR (dB)	39.34	40.04	39.65	39.32	38.99	39.83	40.75	39.80
SSIM	0.964	0.947	0.957	0.960	0.939	0.934	0.931	0.945

The robustness of the algorithm is measured in terms of the NC value between the original watermark and the extracted one, which is already defined in Eq. 7.

4.4 Robustness against attacks

The robustness of the proposed algorithm is validated by using a various kind of attacks, such as noise attacks, filtering attacks, geometric attacks and temporal attacks. The attacking process first decodes the watermarked video, then implements the attack and finally encodes the attacked video in H.264 format with the constant rate factor (CRF) being set to 23. The CRF is used for rate control in most cases of H.264 video compression, and it allows the encoder to attempt to achieve a certain output quality. A subjectively sane range for CRF is 17 to 28, and increasing it by 6 results in roughly half the bit rate. Tables 6 and 7 show the NC values from the watermarked videos under these attacks. Moreover, we test the algorithm's robustness against H.264 compressions by recoding the watermarked videos with different CRF values. The results for compressions are shown in Tables 8 and 9. Below we give detailed descriptions for the attacks used in our experiments.

4.4.1 Image processing attacks

To test the algorithm's robustness against noise attacks, we add Gaussian noise and salt & pepper noise to the watermarked videos. The mean and standard deviation of Gaussian noise are 0 and 10, respectively. The density of salt & pepper noise is 0.05, which means about 5% of the pixels in a frame are affected.

Table 6 Experimental results for the CIF videos under various attacks

	Bridge close	Bridge far	Mad900	Students
Gaussian noise	1.000	1.000	1.000	1.000
Salt & pepper noise	1.000	1.000	0.9428	1.000
Histogram equalization	1.000	1.000	1.000	1.000
Median filtering	1.000	0.7144	1.000	1.000
Rotation (1°)	1.000	1.000	1.000	1.000
Rotation (3°)	1.000	1.000	1.000	1.000
Rotation (5°)	1.000	1.000	1.000	1.000
Rotation (10°)	1.000	0.8528	0.7826	1.000
Scaling (3/4)	1.000	1.000	1.000	0.9354
Scaling (2/3)	1.000	1.000	0.9354	0.8660
Cropping	1.000	1.000	1.000	1.000
Frame averaging	1.000	1.000	1.000	1.000
Frame rate changing (15 fps)	1.000	1.000	1.000	1.000
Frame rate changing (50 fps)	1.000	1.000	1.000	1.000

Table 7 Experimental results for the 1080p videos under various attacks

	Big buck bunny	Elephants dream	Sintel trailer	Sita sings the blues
Gaussian noise	1.000	1.000	1.000	1.000
Salt & pepper noise	0.7974	0.9701	0.9843	0.9354
Histogram equalization	1.000	1.000	1.000	0.9847
Median filtering	0.9847	1.000	1.000	1.000
Rotation (1°)	0.9562	1.000	1.000	1.000
Rotation (3°)	0.9062	1.000	0.9525	0.9062
Rotation (5°)	0.9095	1.000	0.9682	0.9095
Rotation (10°)	0.9428	0.9843	0.9375	0.9263
Scaling (3/4)	0.9847	1.000	1.000	1.000
Scaling (2/3)	0.9398	0.9843	1.000	1.000
Cropping	1.000	1.000	1.000	1.000
Frame averaging	0.9843	0.9843	1.000	1.000
Frame rate changing (15 fps)	1.000	0.9843	1.000	1.000
Frame rate changing (50 fps)	1.000	0.9843	1.000	1.000

We also test the robustness of the proposed algorithm against histogram equalization and median filtering. The histogram equalization is an effective method of contrast adjustment for grayscale images. In experiments, the sum of histogram bins is set to 255 and the Red, Green and Blue components of each frame are separately processed. The median filter is a nonlinear filter used for signal smoothing, and we process the frames of each watermarked video using a median filter of size 5×5 in experiments.

4.4.2 Geometric attacks

Geometric transformations are the most challenging attacks for watermarking algorithms. Here we rotate the watermarked videos by various degrees (1°, 3°, 5° and 10°) and extract the watermark from the rotated videos without performing synchronization. The robustness of the algorithm against scaling attacks is also tested. We downscale each watermarked video to 3/4 and 2/3 of its original size and extract the watermark from the downscaled videos. Finally, we implement cropping attacks by cropping 5% of the pixels from each side of the watermarked videos. It can be seen from Tables 6 and 7 that cropping does not affect the performance of the algorithm but rotation and scaling have slight effects. The cause of the relatively lower performance under rotation and scaling is the interpolation used by these two attacks. Though the performance is affected, the high NC values suggest that the proposed algorithm is robust against rotation and scaling.

Table 8 H.264 compression results for the CIF videos

		Bridge close	Bridge far	Mad900	Students
CRF=23	Bit rate (kbps)	348	239	290	265
	NC	1.000	1.000	1.000	1.000
CRF=28	Bit rate (kbps)	112	48	121	115
	NC	1.000	1.000	1.000	0.8660
CRF=30	Bit rate (kbps)	66	23	88	85
	NC	0.8528	0.1768	0.2041	0.4743

Table 9 H.264 compression results for the 1080p videos

		Big buck bunny	Elephants dream	Sintel trailer	Sita sings the blues
CRF=23	Bit rate (kbps)	3218	4846	3690	5016
	NC	1.000	0.9843	1.000	1.000
CRF=28	Bit rate (kbps)	1959	2070	1427	2267
	NC	0.9525	1.000	1.000	1.000
CRF=33	Bit rate (kbps)	967	1007	729	1346
	NC	0.7016	0.7267	0.9232	0.9520

4.4.3 Temporal attacks

We test the algorithm's robustness to two kinds of temporal attacks, i.e., frame averaging and frame rate changing. For frame averaging, each frame of the given watermarked video is replaced with the average of its five neighboring frames. For frame rate changing, we change the frame rate of each watermarked video to 15 fps and 50 fps. It can be seen from Tables 6 and 7 that these two kinds of temporal attacks rarely affect the performance of the proposed algorithm.

4.4.4 H.264 compression

Tables 8 and 9 show the simulation results for H.264 compression. The compression process takes a watermarked video as input and re-encodes it with different CRF values. The bit rate of each re-compressed video as well as the CRF value is reported. It can be seen that for 1080p videos the watermark can be successfully extracted if the bit rate is around or above 2000 kbps, and for CIF videos this value becomes 120 kbps.

4.5 Discussion

In this subsection we discuss the factors influencing the performance of the algorithm. The first is the bit rate of the watermarked video. It can be seen from Table 7 that the proposed algorithm has a lower performance on Big Buck Bunny (BBB). This is because the watermarked BBB has a bit rate of 2686 kbps, which is the lowest of the four 1080p videos. But in practice, 2686 kbps is a fairly low bit rate for 1080p videos, so the average NC of 0.96 for BBB suggests the proposed algorithm has enough robustness for practical use. Another factor influencing the proposed algorithm is the color range of the video frames. From Table 7, the watermark extracted from Elephants Dream has a high probability of one-bit error. The reason is that the last frames of this video have a lot of large black regions, as shown in Fig. 8. The values of the watermarked pixels of such regions are often negative, so they are clipped to zeros in the result, which can be regarded as a strong attack. This is a common phenomenon in many watermarking algorithms, but fortunately most real videos have a less extreme color range. Lastly we point out that the size or resolution of the original video has almost no influence on the algorithm, since it has quantitatively similar performances on the CIF and 1080p videos.

4.6 Performance comparison

In this subsection, we provide some discussions on our experimental results by comparing them with the results obtained by other two related methods proposed by Faragallah [7] and

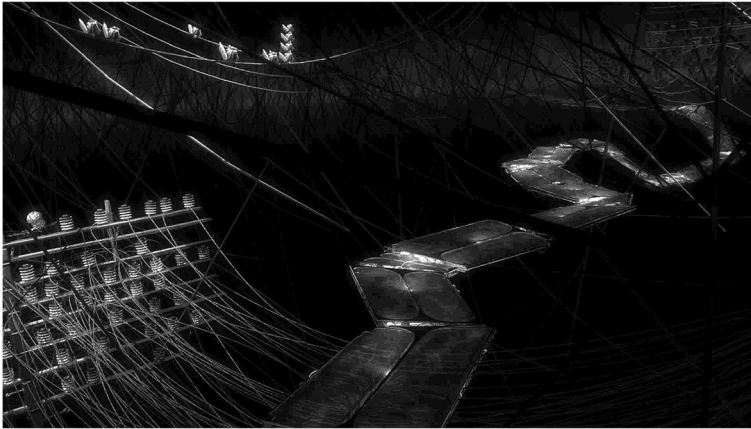


Fig. 8 A frame at the end of elephants dream

Chen et al. [2]. Chen’s algorithm uses the normalized correlation as the robustness measure while Faragallah’s algorithm calculates the bit error rate as well as the normalized correlation. The watermark in [7] is a 54×12 binary image and the watermark in [2] is a pseudo-noise sequence that carries one bit of message. The comparisons are performed on the CIF videos, so the watermark of the proposed algorithm is a 16-bit sequence. The comparison results are shown in Table 10.

For the algorithm in [2], the PSNR values of the watermarked videos are around 40 dB, which are very close to our results. The average NC value of the CIF videos compressed with a bit rate of 1000 kbps is around 0.75, while our algorithm has an NC of 1.0 when the bit rate is above 200 kbps. The algorithm in [7] has a higher average PSNR of 44 dB, and the H.264 compression does not affect its performance. However, it has to be pointed out that the reason for the higher imperceptibility and robustness against compressions is partly due to the semi-blindness of the method, i.e., its extraction process requires the singular vectors of the original frames.

Our scheme has better performance over the other methods in terms of robustness against geometric attacks. The algorithms in [2, 7] need to perform resynchronization to extract the watermark from rotated or scaled videos, but our algorithm does not require to do so. In our experiments the average NC of rotated videos is 0.91 and the corresponding NC values in [2, 7] are 0.88 and 0.84, respectively. For scaling attacks, the average NC of our algorithm is 0.97

Table 10 Comparison between different methods

	Faragallah [7]	Chen et al. [2]	Proposed
Blindness	Semi-blind	Blind	Blind
Geometric resynchronization	Required	Required	Non-required
Robustness against cropping	Non-robust	Non-robust	Robust
PSNR	44.85	40.80	39.59
Compression (1000 kbps)	1.00	0.75	1.00
Rotation	0.88	0.84	0.91
Scaling	0.95	0.85	0.97
Gaussian noise	1.00	0.75	1.00
Salt & pepper noise	1.00	0.84	0.95
Median filtering	0.96	0.95	0.93

and the NC values of [2, 7] are 0.95 and 0.85, respectively. In addition, the two algorithms in [2, 7] cannot deal with cropping attacks, since the spatial synchronization of the frames are destroyed.

For image processing attacks, the proposed algorithm also has a comparable performance. The Gaussian noise rarely affects Faragallah's and our algorithms, but the NC value of [2] is only 0.75. For salt & pepper noise, the average NC of our algorithm is 0.95, the NC value of [2] is 0.84 and the error rate of [7] is 0.0%. In our algorithm the average NC of the videos processed by median filter is 0.93, the corresponding NC in [2] is 0.95 and the error rate in [7] is 4.11%.

5 Conclusion and future work

In this paper, a geometrically robust video watermarking algorithm based on 2-D DFT is proposed. For embedding the watermark, circular templates in DFT domain are transformed into noise masks and added to the video frames in spatial domain. The extraction process is based on the cross-correlation between the Wiener-filtered DFT magnitude and the stretched template sequence in polar coordinates. To improve the fidelity of the watermarked video, we also propose a perceptual model based on local contrast of the frames. The main contributions of this paper can be summarized as follows:

- (1) The DFT template is used to handle the RST attacks on videos. To embed a multi-bit watermark, we design two kinds of templates that represent 0 and 1 respectively and segment the host video into GOPs each of which holds one bit of watermark.
- (2) The linearity of DFT is used to accelerate the frame-by-frame watermarking scheme. In embedding, we avoid the computation of DFT by adding the noise masks in spatial domain. In extraction, we only need to perform a DFT on the average image of each GOP.
- (3) The process of detecting the DFT template in an image is improved. This is also useful for image watermarking algorithms that use the DFT template as a synchronization pattern.

It can be seen from the experiments that the proposed algorithm has a good robustness and imperceptibility. The algorithm can resist general image processing attacks, such as the noise attacks, filtering attacks and contrast adjustment attacks. For geometric attacks, the algorithm also has a good performance and the extraction process does not require synchronization. To demonstrate the practicability of the proposed algorithm, we use some 1080p videos of H.264 format in the experiments and test the algorithm's robustness against H.264 compression. The results show that our algorithm is very robust to the modern compression technique.

One of the shortcomings of the proposed algorithm is its relatively low capacity. Also, the algorithm is not robust against frame dropping attacks. The temporal synchronization can be destroyed by randomly dropping some of the frames. One solution may be to add a third kind of DFT template for temporal synchronization, but this can further decrease the capacity. We will carefully consider these problems in the future work.

Acknowledgments This work is supported by the Alibaba-Zhejiang University Joint Institute of Frontier Technologies.

References

1. Barni M, Bartolini F, Piva A (2001) Improved wavelet-based watermarking through pixel-wise masking. *IEEE Trans Image Process* 10(5):783–791
2. Chen L, Zhao J (2018) Contourlet-based image and video watermarking robust to geometric attacks and compressions. *Multimed Tools Appl* 77(6):7187–7204
3. Chou CH, Li YC (1995) A perceptually tuned subband image coder based on the measure of just-noticeable-distortion profile. *IEEE Trans Circuits Syst Video Technol* 5(6):467–476
4. Cox IJ, Miller ML, Bloom JA, Fridrich J, Kalker T (2009) *Digital watermarking and steganography*, second edn. Morgan Kaufmann Publisher, San Francisco
5. Deng C, Gao X, Li X, Tao D (2010) Local histogram based geometric invariant image watermarking. *Signal Processing* 90(12):3256–3264
6. Dutta T, Gupta HP (2016) A robust watermarking framework for high efficiency video coding (HEVC)-encoded video with blind extraction process. *J Vis Commun Image Represent* 38:29–44
7. Faragallah OS (2013) Efficient video watermarking based on singular value decomposition in the discrete wavelet transform domain. *AEU-INT J Electron* 67(3):189–196
8. Kaehler A, Bradski G (2016) *Learning OpenCV 3: computer vision in C++ with the OpenCV library*. O'Reilly Media, Inc., California
9. Kuan DT, Sawchuk AA, Strand TC, Chavel P (1985) Adaptive noise smoothing filter for images with signal-dependent noise. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 7(2):165–177
10. Liao X, Li K, Yin J (2017) Separable data hiding in encrypted image based on compressive sensing and discrete fourier transform. *Multimed Tools Appl* 76(20):20739–20753
11. Liao X, Guo S, Yin J, Wang H, Sangaiah AK (2018) New cubic reference table based image steganography. *Multimed Tools Appl* 77(8):10033–10050
12. Lim JS (1990) *Two-dimensional signal and image processing*. Prentice Hall PTR, New Jersey
13. Liu Y, Zhao J (2010) A new video watermarking algorithm based on 1D DFT and Radon transform. *Signal Processing* 90(2):626–639
14. Mansouri A, Aznaveh AM, Torkamani-Azar F, Kurugollu F (2010) A low complexity video watermarking in H. 264 compressed domain. *IEEE Transactions on Information Forensics and Security* 5(4):649–657
15. Mukherjee DP, Maitra S, Acton ST (2004) Spatial domain digital watermarking of multimedia objects for buyer authentication. *IEEE Trans Multimed* 6(1):1–15
16. Noorkami M, Mersereau RM (2005) Compressed-domain video watermarking for H. 264. *IEEE International Conference on Image Processing* 2005(2):II–890
17. Pereira S, Pun T (2000) Robust template matching for affine resistant image watermarks. *IEEE Trans Image Process* 9(6):1123–1129
18. Pramila A, Keskinarkaus A, Seppänen T (2007) Multiple domain watermarking for print-scan and JPEG resilient data hiding. *International Workshop on Digital Watermarking*, Springer, Berlin, Heidelberg, pp 279–293
19. Preda RO, Vizireanu DN (2015) Watermarking-based image authentication robust to JPEG compression. *Electronics Letters* 51(23):1873–1875
20. Singh AK, Dave M, Mohan A (2016) Hybrid technique for robust and imperceptible multiple watermarking using medical images. *Multimed Tools Appl* 75(14):8381–8401
21. Solachidis V, Pitas L (2001) Circularly symmetric watermark embedding in 2-D DFT domain. *IEEE Trans Image Process* 10(11):1741–1753
22. Suhail MA, Obaidat MS (2003) Digital watermarking-based DCT and JPEG model. *IEEE Trans Instrum Meas* 52(5):1640–1647
23. Thakkar FN, Srivastava VK (2017) A blind medical image watermarking: DWT-SVD based robust and secure approach for telemedicine applications. *Multimed Tools Appl* 76(3):3669–3697
24. Urvoy M, Goudia D, Atrousseau F (2014) Perceptual DFT watermarking with improved detection and robustness to geometrical distortions. *IEEE Trans Inf Forensics Secur* 9(7):1108–1119
25. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Process* 13(4):600–612
26. Xiph.Org Foundation (2020) Xiph.org Video Test Media [derf's collection]. <https://media.xiph.org/video/derf>. Accessed 20 April 2020