Digital video watermarking robust to geometric attacks and compressions

by

Yan Liu

A thesis
submitted to the Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

Doctorate of Philosophy
in
Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Electrical Engineering and Computer Science (EECS)
University of Ottawa

Abstract

This thesis focuses on video watermarking robust against geometric attacks and video compressions. In addition to the requirements for an image watermarking algorithm, a digital video watermarking algorithm has to be robust against advanced video compressions, frame loss, frame swapping, aspect ratio change, frame rate change, intra- and inter-frame filtering, etc. Video compression, especially, the most efficient compression standard, H.264, and geometric attacks, such as rotation and cropping, frame aspect ratio change, and translation, are considered the most challenging attacks for video watermarking algorithms.

In this thesis, we first review typical watermarking algorithms robust against geometric attacks and video compressions, and point out their advantages and disadvantages. Then, we propose our robust video watermarking algorithms against Rotation, Scaling and Translation (RST) attacks and MPEG-2 compression based on the log-polar mapping and the phase-only filtering method. Rotation or scaling transformation in the spatial domain results in vertical or horizontal shift in the log-polar mapping (LPM) of the magnitude of the Fourier spectrum of the target frame. Translation has no effect in this domain. This method is very robust to RST attacks and MPEG-2 compression. We also demonstrate that this method can be used as a RST parameters detector to work with other watermarking algorithms to improve their robustness to RST attacks.

Furthermore, we propose a new video watermarking algorithm based on the 1D DFT (one-dimensional Discrete Fourier Transform) and 1D projection. This algorithm enhances the robustness to video compression and is able to resist the most advanced

video compression, H.264. The 1D DFT for a video sequence along the temporal domain 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 performance of the proposed algorithm is evaluated by video compression standards MPEG-2 and H.264; geometric attacks such as rotation, translation, and aspect-ratio changes; and other video processing. The most important advantages of this video watermarking algorithm are its simplicity, practicality and robustness.

Contents

List of Tables

List of Figures

List of Acronyms

LPM	Log Polar Mapping
ILPM	Inverse Log Polar Mapping
RST	Rotation, Scaling and Translation
DFT	Discrete Fourier Transform
IDFT	Inverse Discrete Fourier Transform
HVS	Human Visual System
DCT	Discrete Cosine Transform
FMT	Fourier-Mellin Transform
DWT	Discrete Wavelet Transform
LSB	Least Significant Bit
PN	Pseudo Noise
PSNR	Peak Signal-to-Noise Ratio
SNR	Signal-to-Noise Ratio
PCE	Peak-to-Correlation Energy
POF	Phase-Only Filter
BPOF	Binary Phase-Only Filter
HAS	Human Auditory System
2D	Two Dimensional
3D	Three Dimensional
PVEQ	Perceptual Evaluation of Video Quality
SSIM	Structural Similarity

This thesis is dedicated to Rongchun, Xiaolin, Junchen, parents and parents-in-law.

Acknowledgement

I would like to deeply thank my supervisor, Professor Jiying Zhao, for bringing the problem of digital watermarking to me, and for his valuable guidance and feedbacks during every step of my work. I greatly appreciate his patience and confidence in my research ability.

Chapter 1

Introduction

1.1 Digital watermarking

In the past decade, internet works perfectly with distribution of digital data for pictures, music and videos. Although digital data has many advantages over analog data, the rightful ownership of the digital data source is at risk. The copyright protection for digital media becomes an important topic for researchers.

The current encryption technology delivers encrypted contents and a decryption key to those who have purchased legitimate copies of the content. However, after the contents are decrypted, it is still hard to trace illegal reproductions. As a result, digital watermarking has been investigated as a complementary technology.

Fig. 1.1 is a typical watermarking system, which includes a watermark embedder and a watermark detector. The inputs to the watermark embedder are the watermark, the cover media data and the security key. The cover media may or may not be transformed to different domains prior to the watermark embedding. The watermark can be a number sequence or a binary bit sequence. The key is used to enhance the security

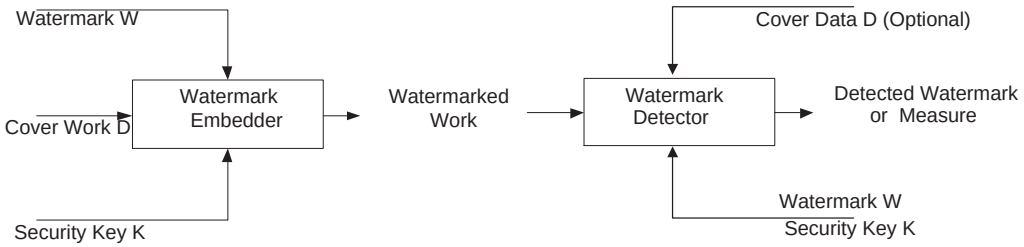


Figure 1.1: A typical watermarking system.

of the whole system. The output of the watermark embedder is the watermarked data.

The inputs to the watermark detector are the watermarked data, the watermark, the security key and, depending on the method, the original data and/or the original watermark. As discussed by [3], a watermark detector includes two-step process. The first step is watermark extraction that applies one or more pre-processes to extract a vector referred to as extracted mark. Then the second step is to determine whether the extracted mark contains the original watermark or not. The second step usually involves with comparing the extracted mark with the original watermark and the result could be some kind of confidence measurement indicating how likely the original watermark is present in the work. For some watermarking algorithms, the extracted mark can be further decoded to get the embedded message for various purposes such as copyright protection.

Suppose that a watermark is defined as W, K is the security key, and D is the host data, which could be image, video, audio, 3D virtual objects, holograph, text or software, etc. In watermarking, an embedding function $e(.)$ takes the watermark W, the host data D, and the security K, as the input parameters, and outputs the watermarked data D'.

$$D' = e(D, W, K) \tag{1.1}$$

The watermark is considered to be robust if it is embedded in a way such that the watermark can survive even the watermarked data D' go through severe distortions. The watermark detection procedure is depicted as follows:

$$W' = d(D', K, W, \ldots) \tag{1.2}$$

where $d(.)$ is the detection function. K and W are the optional inputs for the detection function.

Watermark detection can be thought as watermark extraction when the watermark carries only one bit information that indicates if the original watermark is present in the work or not.

For a typical watermarking system, several requirements should be satisfied:

1. The watermark W' can be detected from D' with or without requiring explicit knowledge of D.

2. D' should be as close to D as possible in most cases.

3. If D' is unmodified, then the detected watermark W' exactly matches W.

4. For robust watermarking, if D' is modified, W' should still match W well to give a clear judgment on the existence of the watermark.

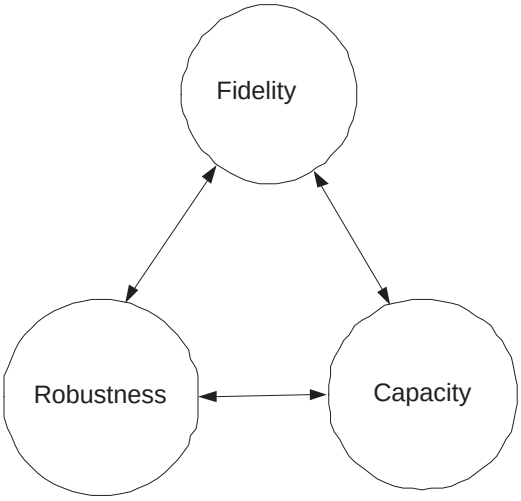


Figure 1.2: The requirements for a robust digital watermarking system.

5. For fragile watermarking, W' will be significantly or totally different from W after even the slightest modification to D'. W' can indicate the possible tampering to D' and give information about the degradation of D'.

1.2 Digital video watermarking

Digital video watermarking technique embeds cryptographic information to the video signals. Ideally, the watermark has to be imperceptible to the users. The watermark extraction algorithm should successfully detect and retrieve the watermark data. Digital video watermarking schemes have to be optimized according to three essential factors, shown in Fig. 1.2:

1. Robustness and security

   Robustness and security are the ability to detect the watermark after attacks. We call a watermark scheme robust if it can resist common signal processing,

such as, rotation, scaling, compression, filtering, noise, aspect ratio change, frame dropping, and frame swapping. Security refers to the ability to resist the attacks from the hostile attackers, such as, collusion attack, which means malicious users merge their knowledge to produce illegal unwatermark data [4].

2. Perceptual fidelity

   Theoretically, the watermark could be perceptible or imperceptible. However, in most applications, an imperceptible watermarking system is preferred. In other words, to keep the good fidelity of watermarked work is required. The fidelity of a watermarking system refers to the perceptual similarity between the original and watermarked versions of the cover work [3].

3. Watermark capability

   Capacity means the maximum amount of information the embedded watermark can carry and those information can be detected reliably for copyright protection and authentication.

As mentioned by O'Ruanaidh and Pun [5], "the smaller is the number of bits of core information or payload contained in a watermark, the greater the chance of it being communicated without error." One way to implement the spread spectrum based watermarking scheme is to generate the watermark as pseudo random sequence, the watermark detection is executed by computing the correlation between the extracted mark and the original watermark. This approach is very robust however the capacity is low (only one bit). A good watermarking algorithm should achieve a good trade-off among these requirements, refer to Fig. 1.2.

1.3 Evaluation of video watermarking

It is extremely difficult to identify the acceptable fidelity because too little is known about human perception. We usually use the objective and subjective measurement to define a threshold. For example, mean squared error (MSE, refer to Equ. (1.3)), signal-to-noise ratio (SNR, refer to Equ. (1.4)) and peak signal-to-noise ratio (PSNR, refer to Equ. (1.5), assume 8-bit pixel depth), are objective measurements. In these equations, c_1 and c_2 are N-dimensional vectors in media space. Just noticeable difference (JND) is the subjective measurement. In the field of psychophysics, a JND is the amount that something must be changed for the difference to be noticeable.

$$D_{mse}(c_1, c_2) = \frac{1}{N}\sum_{i}^{N}(c_1[i] - c_2[i])^2 \tag{1.3}$$

$$D_{snr}(c_1, c_2) = 10\log_{10}\frac{\sum_i^N (c_1[i])^2}{\sum_i^N (c_2[i] - c_1[i])^2} \tag{1.4}$$

$$D_{psnr}(c_1, c_2) = 10\log_{10}\frac{255^2}{\frac{1}{N}\sum_i^N (c_1[i] - c_2[i])^2} \tag{1.5}$$

The above mentioned three measurements are traditionally used for image and video quality. Recently several more complicated and precise matrices are developed. Most popular methods are PEVQ and SSIM.

PEVQ - Perceptual Evaluation of Video Quality, provided by OPTICOM, is a full

reference and intrusive measurement algorithm with the reference of the original sequence for comparison [6]. The algorithm compares each corresponding pixel among the reference and the degraded sequences regarding several factors:

- Spatial and temporal alignment ensures only the corresponding parts from reference and the target signals are compared.
- Perceptual difference alignment only reviews the parts, which can be perceptually noticed by human viewers.
- Types of distortions are classified from the previously calculated indicators.
- The result - the mean opinion score (MOS), formed with the aggregation of above mentioned steps, estimates the affect from loss, jitter, block, jerkiness, blur and distortion.

PEVQ MOS rates the video quality in the range from one (bad) to five (excellent).

SSIM - Structural Similarity based quality measurement is a more direct measurement way to compare the structures of the reference and the degraded signals [7]. The similarity measurement is combined with three components, Luminance comparison, Contrast comparison and Structure comparison. This method is considered as an objective method to evaluate the image quality using the properties of human visual system. The maximum index of SSIM is 1, which means two blocks are exactly the same.

1.4 Applications of video watermarking

Digital video watermarking systems are developed based on the applications. The following applications of watermarking are more common [4]:

1. Copyright protection: It is to identify the copyright ownership. The copyright owner can prove the ownership by detecting the watermark embedded in the host with their private keys. The watermarks used for this purpose are supposed to be very robust against various attacks intended to remove the watermark. This application is quite straightforward. Many researchers focused their research on this topic [8] [9] [10] .

2. Broadcast monitoring: The broadcasted channels need to be tracked to keep the royalties of the copyrighted videos. A computer system could monitor the broadcasted video by simulating a human observer, which is a passive monitoring. The passive monitoring depends on the comparison between the received video with the original signals, which needs the large database storage and may not be available often. Digital video watermarking could provide an active monitoring in an invisible and robust way. The watermark pattern can be embedded as the identification information in the broadcast signals and be detected reliably and interpreted correctly. Real time watermarking allows the live active monitoring for many channels [11][12].

3. Copy control: It is to prevent the unauthorized copies of copyrighted content. Different payment entitles users to have different privilege (play, copy) on the object. It is desirable in some systems to have a copy and usage control mechanism to prevent illegal copy of the content or limit the number of times of copying. A watermark [13] can be used for such purposes.

4. Content authentication: To be able to authenticate the content, any change or tampering with the content should be detected. This can be achieved through "fragile or semi-fragile watermark" which has low robustness to the modifications

to the host video. The semi-fragile watermarking can also serve the purpose of quality measurement. The extracted watermark cannot only tell the possible tampering with the host signals, but also give more information about the degradation of the host, such as PSNR degradation. Many researchers proposed watermarking algorithms for video authentication and tamper detection [14][15][16].

5. Forensic analysis: Digital Rights Management (DRM) systems are designed to protect and enforce copyright control with digital media content such as audio, video or still images [17]. A forensic watermarking is embedded into the master copy of the content and it allows the owner to identify the illegal or unauthorized use and distribution of copyrighted digital work. In digital cinema application, forensic watermark is used to determine the copyright owner, where and when the theft occurs and to prevent such illegal activities [18][19][20].

6. Enhanced coding: Digital watermarking has other functionalities accept for security purpose. It could be used for carrying error correction information after source coding without any header addition [21]. It also be used as a method to hide video or audio into anther video to save bandwidth and enhance synchronization [22] [23].

1.5 Challenges of video watermarking

Digital watermarking has to be robust against various attacks. There are two major categories of attacks for digital video watermarking. One is common signal processing, while the other is hostile video attacks.

The first category refers to a wide range of video processing. We could divide the

common video processing to two subsets. Subset I is video processing which modifies the pixel values of frames but keeps the spatial or temporal synchronization. Subset II is video processing which causes spatial or temporal desynchronization.

Subset I includes many video processings as follows:

1. Digital to analog and analog to digital conversions introduce distortions in the video signal.
2. Data transmission probably introduces additive noise to the signal.
3. Spatial filtering is often used to restore low quality video.
4. Gamma correction is used to increase contrast.
5. Histogram equalization is used to adjust and increase contrast.
6. Lighting change modifies the luminance components.
7. Chrominance resampling (4:4:4, 4:2:2, 4:2:0) is normally a part of video compression to reduce the video size for transmission and storage.
8. Video Compression.

Subset II includes some other video processing to distort the spatial or temporal synchronization:

1. Geometric attacks, such as, rotation, scaling and translation will distort the spatial synchronization. Scaling for video can be explained as frame aspect ratio change.
2. Frame rate change, frame loss, and frame swapping could cause temporal desynchronization.

The second category is classified as hostile video attacks. Collusion could uncover unwatermarked data or watermark pattern itself with the information from a set of malicious users. There are two types of collusion attacks [4]. The type I refers that, when the same watermark is embedded into different data, attackers can estimate the watermark pattern from each watermarked data. The type II is that, when different watermarks are embedded into the different copies of the same data, attackers can produce the unwatermarked data from the linear combination of the different watermarked data.

From the attacks we mentioned in the first category, we consider geometric attacks and video compression, especially, the most advanced video compression standard, H.264, as the most challenging attacks for video watermarking algorithms.

Geometric attacks include rotation, scaling and translation (RST) transform, and other linear or non-linear transformation. In this thesis, we deal with RST attacks. Rotation repositions each pixel of an image along a circular path, while translation transforms each pixel along a straight-line path from one coordinate location to another. Scaling alters the size of the target frame. Very slight of RST transformation could cause detector to lose spatial synchronization and make detection fail.

Video compression reduces the spatial and temporal redundancy information from the large video data while keeping the good perceptual quality. It is a tradeoff among video quality, disk space, and the cost of hardware [24]. Video compression operates on the blocks of neighboring pixels. To reduce spatial redundancy, lossy video compression gives coarse quantization to higher frequency information and finer quantization to lower frequency information because the human visual system is more sensitive to low frequency information. To reduce temporal redundancy, video compression codec only sends the differences among moving blocks. MPEG-2, MPEG-4 and H.264 are

popular video compression standards. H.264/AVC is the latest block-oriented motion-compensation-based codec standard developed by the ITU-T Video Coding Experts Group (VCEG) together with the ISO/IEC Moving Picture Experts Group (MPEG). With the use of H.264, up to 50% of bit rate could be saved.

1.6 Objectives of the thesis

The thesis has the two main objectives:

1. To research and develop video watermarking algorithms robust against RST attacks: RST attacks cause the loss of the spatial synchronization which results in the failure of watermark detection. RST attacks are considered the most challenge for both image and video watermarking algorithms. In this thesis, we will try to research and develop video watermarking algorithms that are robust to RST attacks, by exploring most advanced theories and techniques to provide RST invariance.

2. To research and develop video watermarking algorithms robust against video compressions: Video compression is used to remove redundancy and reduce the size of the video signals. MPEG-2 and H.264 are commonly used video compression standards and H.264 is considered the most efficient video compression technique, which can provide high quality with the low bit rate. It is extremely difficult for a video watermarking algorithm to survive H.264 compression. In this thesis, we will research and develop advanced video watermarking algorithms that are robust to video compressions including H.264 compression, by considering the features of the video compression standards.

1.7 Contributions of the thesis

The contributions of the thesis are summarized as follows:

1. The thesis reviews and evaluates the existing video watermarking algorithms robust against RST transformations and video compressions, by giving the advantages and disadvantages of the existing watermarking algorithms.

2. An RST invariant video watermarking algorithm (LPMPOF) based on log-polar mapping and phase-only filtering method is proposed. Rotation and scaling of video signals in the spatial domain result in the horizontal and vertical shift in log-polar mapping of Fourier magnitude of the same video signals. Embedding watermark in the log-polar mapping domain simplifies the complex rotation and scaling problems into image registration. The proposed phase-only filtering method provides the best discrimination for matching a template and log-polar mapping of Fourier magnitude of the watermarked frame. The optimized embedding and detection method provides the robustness to RST attacks, other video signal processing, and MPEG-2 compression.

3. The integration of LPMPOF with another video watermarking algorithm is demonstrated. The purpose is to improve the robustness of the other algorithm against RST attacks. The role of LPMPOF in the integration is an RST parameters detector. The LPMPOF can also contain a robust watermark. The other video watermarking algorithm can provide a large capacity.

4. A novel video watermarking algorithm based on 1D DFT and 1D projection is proposed. This algorithm is developed with the consideration of the features of H.264 video compression techniques. The 1D DFT of a group of pictures along

temporal direction holds both the spatial and frequency information and it provides an ideal domain to embed the watermark pattern in an invisible and robust way. The embedding method keeps the features of geometric relationship between watermark pattern and frame, which ensures robustness against geometric attacks. The proposed algorithm is robust against geometric attacks and H.264 compression according to the experimental results.

1.8 Publications

Refereed journals:

1. Yan Liu and Jiying Zhao, A new video watermarking algorithm based on 1D DFT and Radon transform, Elsevier Journal: Signal Processing, Vol. 90, Issue 2, pp. 626-239, February 2010.

2. Yan Liu, Dong Zheng, and Jiying Zhao, An image rectification scheme and its applications in RST invariant digital image watermarking, Springer Journal: Multimedia Tools and Applications, Vol. 34, No. 1, pp. 57-84, July 2007.

3. Dong Zheng, Yan Liu, Jiying Zhao, and Abdulmotaleb El Saddik, A Survey of RST Invariant Image Watermarking Algorithms, ACM Computing Surveys, Vol. 39, No. 2, Article 5, pp. 1-91, June 2007.

4. Dong Zheng, Yan Liu, and Jiying Zhao, RST invariant digital image watermarking based on a new phase-only filtering method, Elsevier Journal: Signal Processing, Vol.85, No.12, pp.2354-2370, December 2005.

5. Yan Liu, Xiangsheng Wu, Dong Zheng, Jiying Zhao, and Jianping Yao, Digital Watermarking and Its Applications in Electric Power Systems, Electric Power

Information Technology (ISSN 1672-4844, CN 11-5060/TK), Vol.3, No.6, pp. 91-92, 2005.

6. Yan Liu, Xiangsheng Wu, Dong Zheng, Jiying Zhao, and Jianping Yao, Phase Information in RST Invariant Image Watermarking, the Proceedings of the CSEE (Chinese Society of Electrical Engineering), Vol.25, No.10, pp. 89-96, 2005.

7. Yan Liu, Dong Zheng, and Jiying Zhao, A Rectification Scheme for RST Invariant Image Watermarking, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Special Section on Cryptography and Information Security, Vol. E88-A, No.1, pp. 314-318, January 2005. LETTER.

Refereed proceedings:

1. Yan Liu and Jiying Zhao, RST invariant video watermarking based on log-polar mapping and phase-only filtering, ICME 2010, Workshop on Content Protection & Forensics, Singapore, July 19-23, 2010.

2. Yan Liu and Jiying Zhao, RST invariant video watermarking based on 1D DFT and Radon transform, The 5th IET Visual Information Engineering 2008 Conference (VIE'08), Xian, China, July 29 August 1, 2008.

3. Yan Liu, Dali Wang, and Jiying Zhao, Video watermarking based on scene detection and 3D DFT, the IASTED International Conference on Circuits, Signals and Systems (CSS 2007), Banff, Canada, pp. 124-130, July 02-04, 2007.

4. Yan Liu and Jiying Zhao, A Robust Image Watermarking Method Based on Adaptive Feature Points Detection, Proc. International Conference on Communications, Circuits and Systems (IEEE), Vol. I, Track 05, Guilin, China, pp.49-53, June 25-28, 2006.

5. Dong Zhang, Yan Liu, and Jiying Zhao, A survey of RST Invariant Image Watermarking Algorithms, Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) 2006, Ottawa, Ontario, Canada, pp.2055-2058, May 7-10, 2006.

6. Luc Lamarche, Yan Liu, and Jiying Zhao, Flaw in SVD based Watermarking, Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) 2006, Ottawa, Ontario, Canada, pp.2051-2054, May 7-10, 2006.

7. Yan Liu and Jiying Zhao, A Robust RST Invariant Image Watermarking Method Based on Locally Detected Features , HAVE 2005 - IEEE International Workshop on Haptic Audio Visual Environments and their Applications Ottawa, Ontario, Canada, pp. 133-138, 12 October, 2005.

8. Dong Zheng, Yan Liu, and Jiying Zhao, RST Invariant Digital Image Watermarking Based on a New Phase-Only Filtering Method, Proceedings of 7th International Conference on Signal Processing (ICSP04, IEEE, CIE, IEE), Beijing, China, pp.25-28, Aug 31-Sep 4, 2004.

9. Yan Liu and Jiying Zhao, Rotation, Scaling, Translation Invariant Image Watermarking Based on Radon Transform, IEEE Canadian Conference on Computer and Robot Vision (CRV2004), London, Ontario, Canada, pp.225-232, May 17-19 2004.

10. Yan Liu and Jiying Zhao, A rectification scheme for RST invariant image watermarking, IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) 2004, Niagara Falls, Ontario, Canada, pp.527-530, May 2-5, 2004.

11. Yan Liu and Jiying Zhao, A New Filtering Method for RST Invariant Image Watermarking, IEEE International Workshop on Haptic, Audio and Visual Environments and their Applications, Ottawa, Ontario, Canada, pp. 101-106, 20-21 September 2003.

1.9 Thesis structure

The rest of the thesis is organized as follows. In Chapter 2, we introduce the fundamental theories and techniques used in this thesis. In Chapter 3, we review image and video watermarking algorithms robust to geometric attacks and video compressions. In Chapter 4, we propose our RST invariant video watermarking algorithm based on log-polar mapping and phase-only filtering method. In Chapter 5, we use RST invariant video watermarking algorithm as a RST parameters detector to work with other video watermarking algorithms. In Chapter 6, we present another novel video watermarking algorithm based on 1D DFT and 1D projection. Finally, in Chapter 7, we conclude the thesis and give some suggestions and ideas for future research work.

Chapter 2

Fundamental theories and techniques

In this chapter, we introduce fundamental theories and techniques used in geometrical resilient image and video watermarking algorithms and video compressions, which are needed to understand the content of the thesis.

2.1 Geometric transformations

First, we model geometric transformations for images and video frames. For image, we define Rotation, Scaling, and Translation (RST) transformations. For video, we define frame aspect ratio change.

2.1.1 Translation

A translation (or shift) is applied to an image or a frame in video by repositioning it along a straight-line path from one coordinate location to another [25]. We translate

a two-dimensional point by adding translation distances, x_0 and y_0, to the original coordinate position (x, y) to move the point to a new position (x', y').

$$\begin{cases} x' &= x + x_0 \\ y' &= y + y_0 \end{cases} \tag{2.1}$$

The translation distance pair (x_0, y_0) is called a translation vector or shift vector.

2.1.2 Rotation

A two-dimensional rotation is applied to an image or a frame in video by repositioning it along a circular path in the xy plane. We obtain the transformation equations for rotating a point at (x, y) through an angle α about the origin counterclockwise:

$$\begin{cases} x' &= x\cos\alpha + y\sin\alpha \\ y' &= -x\sin\alpha + y\cos\alpha \end{cases} \tag{2.2}$$

2.1.3 Scaling

A scaling transformation alters the size of an image. We obtain the transformation equations by multiplying the coordinate values (x, y) by scaling factors σ_x and σ_y to produce the transformed coordinates (x', y'):

$$\begin{cases} x' &= x \cdot \sigma_x \\ y' &= y \cdot \sigma_y \end{cases} \tag{2.3}$$

Scaling factor σ_x scales images in the x direction, while σ_y scales in the y direction. When σ_x and σ_y are assigned the same value, an uniform scaling maintains the relative image proportions.

2.1.4 Frame aspect ratio change

For video applications, scaling could be explained as frame aspect ratio change. Depending on the applications, digital video formats are summarized in Table 2.1 [26]. The popular video frame aspect ratios are 4:3, 11:9, 16:9, etc. Frame aspect ratio change, here, is the aspect ratio conversion between any two of aspect ratios by using interpolation.

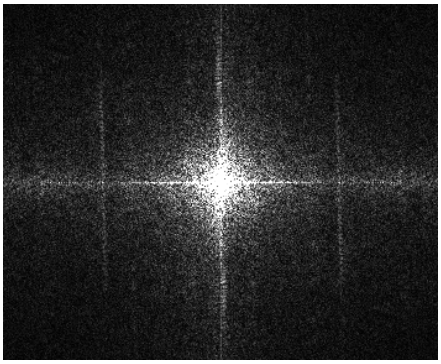
Table 2.1: Video format for different applications

Applications	Video format	Y size
HDTV	SMPTE 296M	1280 × 720
HDTV	SMPTE 274M	1920 × 1080
Video production	BT.601	720 × 480/576
High-quality video distribution (DVD, SDTV)	BT.601	720 × 480/576
Intermediate-quality video distribution (VCD, WWW)	SIF	352 × 240/288
Video conference over ISDN/Internet	CIF	352 × 288
Video telephony over wired/wireless modem	QCIF	176 × 144
Closed Circuit Television (CCTV)	4CIF	704 × 576
High definition video conferencing	4SIF	704 × 480

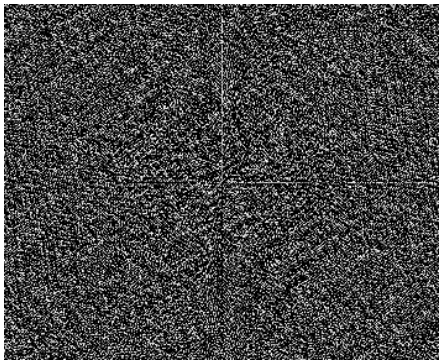
2.2 Fourier-Mellin transforming method

In the previous section, we introduced the mathematic model for RST attacks for image and video signals in the spatial domain. These attacks will make watermark detector lose track of the watermark position in the most applications and make detection fail as a result. Fourier-Mellin transform (FMT) can build a domain, in which rotation, scaling and translation have no effect at all. Fourier-Mellin transforming method includes 2D discrete Fourier transform, log-polar mapping on Fourier magnitude, and FMT on log-polar domain.

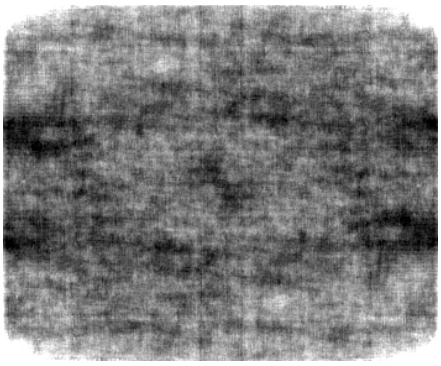
(a) One frame in video "mobile".



(b) The magnitude spectrum of the Fourier transform of (a).



(c) The phase spectrum of the Fourier transform of (a).



(d) The reconstructed frame using only the amplitude spectrum.

(e) The reconstructed frame using only the phase spectrum.

Figure 2.1: DFT and inverse DFT of one frame in video "mobile".

2.2.1 2D discrete Fourier transform

The DFT (Discrete Fourier Transform) of an image is two dimensional (2D). The 2D DFT of an image $f(x, y)$ of size $N_1 \times N_2$ and the corresponding IDFT (Inverse DFT) are defined as follows [27]:

$$F(u,v) = \frac{1}{N_1 N_2} \sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} f(x,y) e^{-j2\pi(ux/N_1+vy/N_2)} \tag{2.4}$$

$$f(x,y) = \sum_{u=0}^{N_1-1} \sum_{v=0}^{N_2-1} F(u,v) e^{j2\pi(ux/N_1+vy/N_2)} \tag{2.5}$$

The Fourier transform $F(u, v)$ is a complex function. Each function value has a real part $R(u, v)$ and an imaginary part $I(u, v)$, at each frequency (u, v) of the frequency spectrum:

$$F(u,v) = R(u,v) + jI(u,v) \tag{2.6}$$

where $j = \sqrt{-1}$. This can be expressed alternately using the exponential form as:

$$F(u,v) = |F(u,v)| e^{j\phi(u,v)} \tag{2.7}$$

where $|F(u, v)|$ is the magnitude of the Fourier transform and $\phi(u, v)$ is the phase angle. The square of the magnitude is equal to the amount of energy or power at each frequency of the image and is defined as:

$$|F(u,v)|^2 = R^2(u,v) + I^2(u,v) \tag{2.8}$$

The phase angle describes the amount of phase shift as each frequency and is defined as:

$$\phi(u,v) = tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right] \tag{2.9}$$

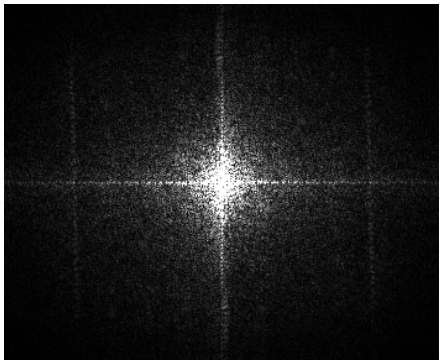
It is understood that the phase information is considerably more important than the amplitude information in preserving the visual intelligibility of the picture. Fourier synthesis of the structure from only the amplitude of the diffraction with zero phases does not reconstruct the correct atomic arrangement, whereas reconstruction from the phase data with unity magnitude does [28]. DFT and inverse DFT processing of one frame in video "mobile" is shown in Fig. 2.1. Fig. 2.1 (e) clearly shows that the frame reconstructed from only the phase information closely resembles the original frame, while the reconstructed frame from only the amplitude information does not, refer to Fig. 2.1 (d).

The relationship between an original image $i_0(x,y)$ and a rotated, scaled, and translated version of the image, $i_1(x,y)$, is shown as follows [1][5]:

$$i_1(x,y) = i_0(\sigma(x\cos\alpha + y\sin\alpha) - x_0, \sigma(-x\sin\alpha + y\cos\alpha) - y_0) \tag{2.10}$$

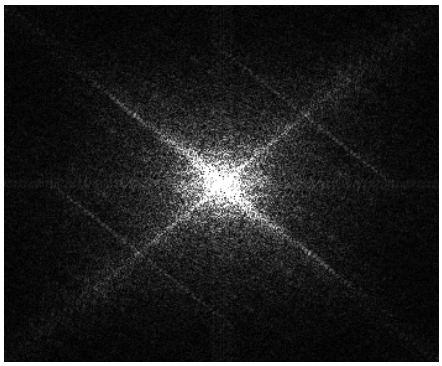
where the RST parameters are α, σ and (x_0, y_0), respectively.

(a) Frame in "mobile" scaled by 80%.



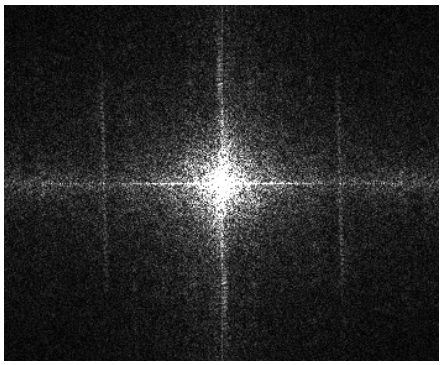
(b) The magnitude spectrum of the scaled frame in (a).

(c) Frame in "mobile" rotated by 45°.



(d) The magnitude spectrum of the rotated frame in (c).

(e) Frame in "mobile" circularly translated by 50 pixels.



(f) The magnitude spectrum of the translated frame in (e).

Figure 2.2: Properties of Fourier transform.

The Fourier transform of $i_1(x, y)$ is $I_1(u, v)$, the magnitude of which is given by [1][29]:

$$|I_1(u,v)| = |\sigma|^{-2}|I_0(\sigma^{-1}(u\cos\alpha + v\sin\alpha), \sigma^{-1}(-u\sin\alpha + v\cos\alpha))| \quad (2.11)$$

As shown in Equ. (2.11), the magnitude spectrum is independent of the translational parameters (x_0, y_0), which is the translation property of the Fourier transform [30]. When an image is rotated in the spatial domain by angle α, its magnitude spectrum will be rotated by the same angle α. Scaling in the spatial domain can cause an inverse scaling in the frequency domain. Fig. 2.2 illustrates all these properties.

2.2.2 Log-polar mapping

If we apply log-polar mapping to the Fourier magnitude of an image, we can rewrite Equ. (2.11) by using log-polar coordinates:

$$\begin{cases} u = e^{\rho}\cos\theta \\ v = e^{\rho}\sin\theta \end{cases} \quad (2.12)$$

where $\rho \in \Re^2$ and $0 \leq \theta < 2\pi$. Then the magnitude of the Fourier spectrum can be written as [1][5]:

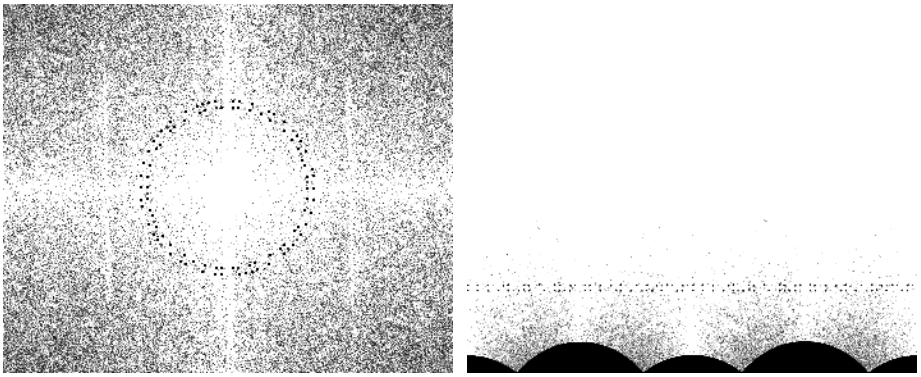
$$|I_1(u,v)| = |\sigma|^{-2}|I_0(\sigma^{-1}e^{\rho}\cos(\theta - \alpha), \sigma^{-1}e^{\rho}\sin(\theta - \alpha))| \quad (2.13)$$

or

$$|I_1(\rho, \theta)| = |\sigma|^{-2} |I_0(\rho - \ln \sigma, \theta - \alpha)| \tag{2.14}$$

Equ. (2.14) demonstrates that the magnitude of the log-polar spectrum is scaled by $|\sigma|^{-2}$, that image scaling results in a translational shift of $\ln \sigma$ along the log-radius ρ axis, that image rotation results in a cyclical shift of α along the angle θ axis, and that image translation has no effect in the LPM domain.

As shown in Fig. 2.3 (a), the watermark data are embedded in two concentric circles in the Fourier magnitude of the target image. In LPM domain, these two circles are transformed into two parallel lines, refer to Fig. 2.3 (b). The vertical and horizontal axes in Fig. 2.3 (b) are respectively log-radius ρ and angle θ. The scaling of image in the spatial domain will result in the watermark circles expanding or shrinking in the Fourier magnitude and a translational shift along the log-radius ρ axis in the LPM domain. The rotation of image in the spatial domain will result in the watermark circle's rotation in the Fourier magnitude and a cyclical shift along the angle θ axis in the LPM domain.



(a) Fourier magnitude of watermarked frame. (b) LPM transform of (a).

Figure 2.3: Fourier magnitude of an watermarked frame and its LPM transform.

The inverse log-polar mapping (ILPM) is defined as:

$$\begin{cases} \rho &= \ln(\sqrt{x^2+y^2}) \\ \theta &= \tan^{-1}(\frac{y}{x}) \end{cases} \tag{2.15}$$

However, ILPM deteriorates the image quality to an unacceptable level due to the implementation difficulty.

2.2.3 Fourier-Mellin transform

According to the translation property of the Fourier transform, the Fourier transforms of I_1 and I_0 in Equ. (2.13) are related by

$$F_1(\omega_\rho, \omega_\theta) = |\sigma|^{-2} e^{-j(\omega_\rho \cdot \ln\sigma + \omega_\theta \cdot \alpha)} F_0(\omega_\rho, \omega_\theta) \tag{2.16}$$

The Fourier magnitude of the two LPM mappings is related by

$$|F_1(\omega_\rho, \omega_\theta)| = |\sigma|^{-2} |F_0(\omega_\rho, \omega_\theta)| \tag{2.17}$$

where F_1 and F_0 are respectively the DFT of I_1 and I_0.

The phase difference between the two LPM mappings is directly related to their displacement, given by $e^{j(\omega_\rho \cdot \ln\sigma + \omega_\theta \cdot \alpha)}$.

Equ. (2.17) is equivalent to computing the Fourier-Mellin transform [5]. Equ. (2.17) demonstrates that the magnitude of Fourier-Mellin spectrum is scaled by $|\sigma|^{-2}$, which is caused by scaling transform, and is invariant to rotation and translation. If normalized correlation is used to eliminate the effect of $|\sigma|^{-2}$, Fourier-Mellin transform is truly invariant to RST.

2.3 Radon transform

The Radon transform represents an image as a collection of projections along various directions [31]. It is used in areas ranging from seismology to computer vision. Projections can be computed along any angle q. 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 as Equ. (2.18), shown in Fig. 2.4.

$$R_q(x') = \int_{-\infty}^{\infty} f(x' \cos q - y' \sin q, x' \sin q + y' \cos q) dy' \tag{2.18}$$

where

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos q & \sin q \\ -\sin q & \cos q \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.19}$$

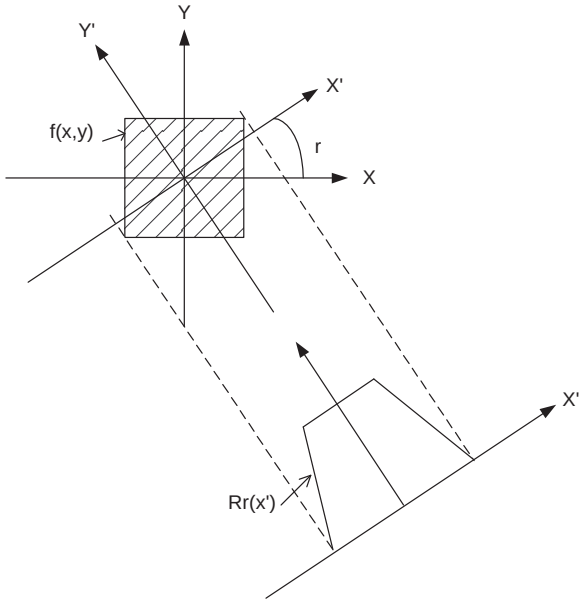
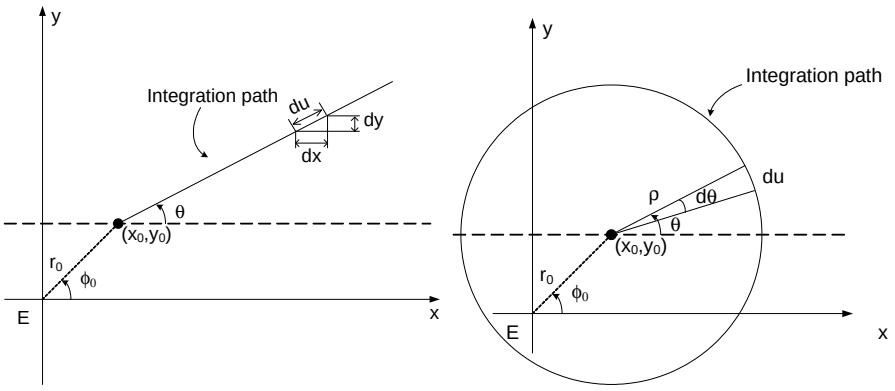


Figure 2.4: Radon transform [1].



(a) Radial Integration Transform (RIT). (b) Circular Integration Transform (CIT).

Figure 2.5: Generalized Radon transforms.

We define $g(\theta)$ to be a 1-D projection of $|I(\rho, \theta)|$ such that [1]:

$$g(\theta) = \sum_j |I(\rho_j, \theta)| \tag{2.20}$$

$g(\theta)$ is invariant to both translation and scaling. Rotations result in a circular shift of the values of $g(\theta)$ [1].

The two one-dimensional generalized Radon transforms were introduced in [32], as shown in Fig. 2.5. The Radial Integration Transform (RIT) of a function $f(x, y)$ is defined as the integral of $f(x, y)$ along a straight line that begins from the origin $f(x_0, y_0)$ and has angle θ with respect to the horizontal axis, as shown in Fig. 2.5 (a). The RIT is defined as follows:

$$R_f(\theta) = \int_0^{+\infty} f(x_0 + ucos\theta, y_0 + usin\theta)du \tag{2.21}$$

If the image is rotated by a certain degree, the RIT of the rotated image will be circularly shifted.

The Circular Integration Transform (CIT) of a function $f(x, y)$ is the integral of $f(x, y)$ along a circle with center $f(x_0, y_0)$ and radius ρ (see Fig. 2.5 (b)). The CIT is given by the following equation:

$$C_f(\rho) = \int_0^{2\pi} f(x_0 + \rho cos\theta, y_0 + \rho sin\theta)\rho d\theta \tag{2.22}$$

The CIT of the scaled image will be scaled the same amount as the image.

So the RIT is independent of scaling, and rotation only results in a shift of the RIT. Similarly, the CIT is independent of rotation, and the CIT is scaled when the image is scaled. Using these properties, we can detect the rotation and scaling the image has undergone by using the CIT and RIT.

2.4 Circular harmonic functions

Circular harmonic expansion is another method used for shift, scaling and rotation invariant pattern recognition [33][34][35]. The circular harmonic function (CHF) is useful in representing the rotational property of an image, which can be expressed in polar coordinates with period 2π in angle and thus can be expressed in terms of a Fourier series expansion in angle [36]. By taking the single harmonic, a circular harmonic filter is invariant to rotation. As with CHFs for rotation invariance, the radial harmonic filters (RHF) decomposes the object into a set of logarithmic radial harmonics. By taking the single harmonic, a radial harmonic filter function is invariant to the shift and scaling.

Let $f(x, y)$ denote the reference image in Cartesian coordinates, we can transform

$f(x, y)$ into polar coordinates $f(r, \theta)$. Because $f(r, \theta)$ is periodic in θ with period of 2π, we can use a Fourier series expansion in θ as follows [33]:

$$f(r, \theta) = \sum_k f_k(r) e^{jk\theta} \tag{2.23}$$

$$f_k(r) = \frac{1}{2\pi} \int_0^{2\pi} f(r, \theta) e^{-jk\theta} d\theta \tag{2.24}$$

where $f_k(r)$ is the k-th circular harmonic function (CHF) of $f(x, y)$.

Let $h(x, y)$ represent a correlation filter in Cartesian coordinates, a CHF decomposition is as follows:

$$h(r, \theta) = \sum_k h_k(r) e^{jk\theta} \tag{2.25}$$

where

$$h_k(r) = \frac{1}{2\pi} \int_0^{2\pi} h(r, \theta) e^{-jk\theta} d\theta \tag{2.26}$$

Then, the correlation function between $f(x, y)$ and $h(x, y)$ is shown in Equ. (2.27).

$$\begin{aligned} c &= \int\int f(x, y) h^*(x, y) dx dy \\ &= \int_0^{2\pi} d\theta \int_0^{\infty} r dr f(r, \theta) h^*(r, \theta) \\ &= \int_0^{\infty} r dr \int_0^{2\pi} \left[\sum_k f_k(r) e^{jk\theta} \cdot \sum_l h_l^*(r) e^{-jl\theta} \right] d\theta \end{aligned} \tag{2.27}$$

Because $\int_0^{2\pi} e^{j(k-l)\theta} d\theta$ is zero when $k \neq l$, the above equation can be expressed as:

$$c = \sum_{k=-\infty}^{\infty} C_k \tag{2.28}$$

where

$$C_k = 2\pi \int_0^{\infty} f_k(r) h_k^*(r) r dr \tag{2.29}$$

When the input image is rotated by the angle ϕ in the clockwise direction, the correlation function is given as follows:

$$c(\phi) = \sum_{k=-\infty}^{\infty} C_k e^{jk\phi} \tag{2.30}$$

If we only use a single circular harmonic

$$\begin{cases} f_s(r,\theta) = f_k(r) e^{jk\theta} \\ h_s(r,\theta) = h_k(r) e^{jk\theta} \end{cases} \tag{2.31}$$

then, the correlation in Equ. (2.27) is expressed as

$$c_s(\phi) = C_k e^{jk\phi} \tag{2.32}$$

The output's center intensity is a constant as shown in the Equ. (2.32). It is invariant to the rotation of the image. The same strategy can be used for RHF.

2.5 Moments

For a 2-D continuous function $f(x, y)$, the moment of order $(p + q)$ is defined as [27]:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \tag{2.33}$$

for $p, q = 0, 1, 2, \ldots$.

The central moments are defined as

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \overline{x})^p (y - \overline{y})^q f(x, y) dx dy \tag{2.34}$$

where $\overline{x} = \frac{m_{10}}{m_{00}}$ and $\overline{y} = \frac{m_{01}}{m_{00}}$

If $f(x, y)$ is a digital image, then Equ. (2.34) becomes

$$\mu_{pq} = \sum_{x} \sum_{y} (x - \overline{x})^p (y - \overline{y})^q f(x, y) \tag{2.35}$$

A set of seven invariant moments can be derived from the second and third moments [27][37].

The normalized central moments, denoted η_{pq}, are defined as

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \tag{2.36}$$

where

$$\gamma = \frac{p + q}{2} + 1 \tag{2.37}$$

for $p + q = 2, 3, \ldots$.

$$\phi_1 = \eta_{20} + \eta_{02} \tag{2.38}$$

$$\phi_2 = (\eta_{20} + \eta_{02})^2 + 4\eta_{11}^2 \tag{2.39}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{2.40}$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{2.41}$$

$$\begin{aligned} \phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ &+ (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \tag{2.42}$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \tag{2.43}$$

$$\begin{aligned} \phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ &+ (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \tag{2.44}$$

This set of moments is invariant to rotation, scaling, and translation. In other words, all images, which are under rotation, scaling or translation of a same image, will accurately produce the same numerical result for above seven equations.

2.6 Harris detector

Feature points detectors find salient points in natural images. Normally, these points are located near corners and edges of the image. The Harris corner detector is one feature point detectors, developed for 3D reconstruction [38]. It extracts the corner points of an image. A Harris corner detector first calculates the horizonal and the vertical gradients of an image, G_x and G_y. Then, two gradient images are filtered by a low-pass filter to get G'_x and G'_y. Then, the shape matrix M is formed for each pixel

[39]:

$$M(i,j) = \begin{bmatrix} \sum_{m,n}(G'_x(m,n))^2 & \sum_{m,n} G'_x(m,n)G'_y(m,n) \\ \sum_{m,n} G'_x(m,n)G'_y(m,n) & \sum_{m,n}(G'_y(m,n))^2 \end{bmatrix} \tag{2.45}$$

where (m,n) represents all pixel positions of a window area centered at the pixel (i,j). By using $M(i,j)$, the Harris corner detector's output for each image pixel is based on the trace and determinant of M [39]:

$$H(i,j) = det(M(i,j)) - k \cdot trace(M(i,j)) \tag{2.46}$$

where k is an arbitrary constant. Feature points extraction is achieved by searching for the response $H(i,j)$ larger than a threshold η.

In above sections, some basic techniques are introduced and they are widely used in many RST invariant image/video watermarking algorithms, which will be reviewed in the following chapter.

2.7 Video compressions

In this section, we will discuss the other challenge for video watermarking algorithms, which is video compression. Video compression removes the spatial and temporal redundancy information from large video data while keeping well perceptual quality. It is a tradeoff among video quality, disk space, and the cost of hardware [24]. Video compression operates on the blocks of neighboring pixels. The main idea of video compression to remove spatial redundancy is to remove higher frequency information and keep lower frequency information because the human visual system is more sensitive to

low frequency information. To remove temporal redundancy, video compression codec only sends the differences among moving blocks.

Digital video compression standards were developed for different applications [2]. Video coding standard H.261 and H.263 are defined for interactive video communication. H.323 and H.324 are defined for audio-visual communications. Moving Picture Expert Group (MPEG) defined MPEG-1, MPEG-2 and MPEG-4 for entertainment and digital TV. H.264/AVC, the part 10 of MPEG-4, is the most efficient video compression standard so far. In this section, we will first introduce Discrete cosine transform (DCT), which is used widely in source coding. Then, we will introduce video compression standard MPEG-2, MPEG-4 and H.264.

2.7.1 Discrete cosine transform

The 2D DCT of an image $f(x, y)$ with the size of $N_1 \times N_2$ and its inverse transform can be defined as follows [2]:

$$F(u,v) = \sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} f(x,y) cos\Big(\frac{\pi}{N_1}(x+\frac{1}{2})u\Big) cos\Big(\frac{\pi}{N_2}(y+\frac{1}{2})v\Big) \quad (2.47)$$

$$f(x,y) = \sqrt{\frac{2}{N_1 N_2}} \sum_{u=0}^{N_1-1} \sum_{v=0}^{N_2-1} F(u,v) cos\Big(\frac{\pi}{N_1}(u+\frac{1}{2})x\Big) cos\Big(\frac{\pi}{N_2}(v+\frac{1}{2})y\Big) \quad (2.48)$$

Different with DFT, DCT coefficients are real values, different from increasing frequency in a sinusoidal pattern . The lowest coefficient is known as DC coefficient, which locates the upper left corner of DCT transform of an image. The others are AC coefficients with the increasingly higher frequencies. Fig. 2.6 shows the 2D DCT basis functions of an 8×8 matrix.

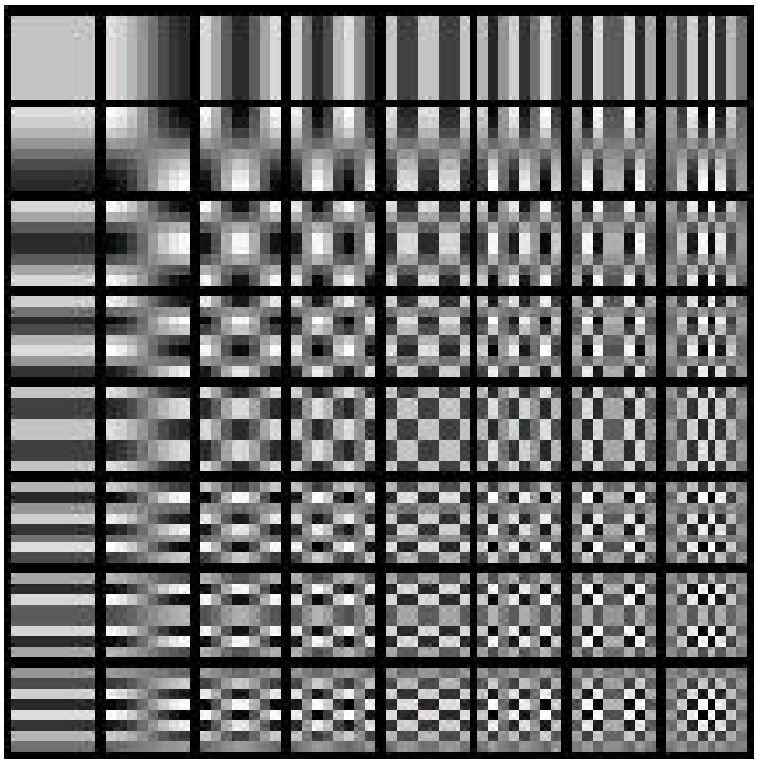


Figure 2.6: 64 basis functions of an 8×8 matrix [2].

2.7.2 MPEG-2

MPEG-2 is widely used for digital TV signals, either SDTV or HDTV broadcasting and communications. It supports both interlaced and progressive video. It allows three options for chrominance format, 4:2:0, 4:2:2 and 4:4:4. The 4:2:0 chrominance format shows that three quarters of the chrominance values have been deleted. The 4:2:2 chrominance format shows that half the chrominance values have been deleted. If no chrominance values have been deleted, the chrominance format is 4:4:4. MPEG-2 specifies that the raw frames be compressed into three kinds of frames: intra-coded frames (I-frame), predictively coded frames (P-frames), and bidirectionally predictively coded frames (B-frames).

I-frame is compressed as a single frame, similar as JPEG compression for images. It does not depend on the data in the preceding or the following frames. Compression of I-frames removes the spatial redundancy according to human visual system. The compression steps can be described as follows:

1. Divide the I-frame into 8×8 blocks.
2. Transform each block with Discrete Cosine Transform (DCT) into DCT frequency domain.
3. Quantize the DCT coefficients with a quantization matrix. In this way, higher frequency components are suppressed.
4. Zigzag the matrix to convert the coefficients into a vector.
5. Apply run-length coding to remove the consecutive zeros.
6. Apply Huffman or arithmetic coding to reduce the matrix to a smaller array of numbers.

P-frames are compressed more than I-frames. Not only the spatial redundancy, but also temporal redundancy will be considered. Each P-frame has a previous I-frame or P-frame as the reference frame. P-frame being compressed is divided into 16×16 macroblocks (MB). The reference frame is searched to find the best matches of the macroblock, being compressed. If a perfect match is found, the offset is encoded as "motion vector". If a similar but not perfect match is found, the difference between two MBs, so called "residual", is appended to the motion vector and sent to the receiver. If no match is found for particular MB, then, this MB is compressed as an I-frame MB.

The processing of B-frames is similar to that of P-frames accept that B-frames use two reference frames, the preceding frame and the following frame. Therefore, B-frames provide even more compression than P-frames. B-frames will never be reference frames in MPEG-2 compression standard.

2.7.3 MPEG-4

MPEG-4 standard is an object-based coding for audio, video and graphics. It provides traditional functionality as well as advanced functionalities, such as, interactivity with individual objects, scalability of contents, and error resilience. MPEG-4 is divided into several parts. The key parts to be aware of are MPEG-4 part 2 and part 10. Part 2 presents a compression codec for visual data (video, still textures, synthetic images, etc.). One of the many "profiles" in Part 2 is the Advanced Simple Profile (ASP). Part 10, Advanced Video Coding (AVC) is a codec for video signals and is technically identical to the ITU-T H.264 standard [24].

One of the important functionality of MPEG-4 is object-based coding. A scene consists of several video objects (VOs). Each VO has the three dimensions (2-D plus time). According to the spatial and temporal positions of a VO, it can be I-VO, P-VO or B-VO. Encoding and decoding work on separate VOs.

Comparing with MPEG-1 and MPEG-2, MPEG-4 increases coding efficiency with following tools:

1. Prediction of the current DC component from the previous block is added.

2. The selected block for DC prediction is also used for predicting one line of AC coefficients. AC prediction does not work well for block with coarse texture, diagonal edges or horizontal or vertical edges.

3. Four motion vectors for a MB are allowed.

4. The motion vector range is unrestricted and it could be as wide as ±2048 pixels.

5. Global Motion Compensation is allowed to improve picture quality due to global motion, such as, camera motion, camera zoom or moving object. Global motion is compensated according to the eight-parameter motion model in Equ. (2.49).

$$\begin{cases} x' &= \frac{ax+by+c}{gx+hy+1} \\ y' &= \frac{dx+ey+f}{gx+hy+1} \end{cases} \tag{2.49}$$

6. Quarter pixel motion compensation is allowed.

2.7.4 H.264

H.264/AVC is the latest block-oriented motion-compensation-based codec standard developed by the ITU-T Video Coding Experts Group (VCEG) together with the ISO/IEC Moving Picture Experts Group (MPEG). H.264 is the most efficient video compression standard with good quality and lower bit rates than previous standards [40].

Some new features of H.264 improve the performance the compression in several aspects, such as, multi-picture inter-picture prediction, spatial prediction, lossless macroblock coding, flexible interlaced-scan video coding, new transform design, new quantization design, in-loop deblocking filter design, an entropy coding design, loss resilience features, auxiliary pictures, support of monochrome, and picture order count. We list some key features here:

1. Allow up to 16 reference frames (or 32 reference fields) during inter-picture pre-

diction.

2. Variable block-size motion compensation is allowed. The luminance prediction block sizes include 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, and 4×4. The chrominance prediction block sizes are correspondingly smaller according to the chroma subsampling in use.

3. Up to 32 motion vectors per macroblock could be used and the motion vectors can point to different reference pictures.

4. Quarter-pixel motion is derived by linear interpolation of the halfpel values and Quarter-pixel precision for motion compensation is enabled for precise description of the displacements of moving areas.

5. Spatial prediction for edges from reference blocks for "intra" coding is allowed.

6. The motion-compensated prediction signal is allowed to be weighted by amounts to improve coding efficiency for scenes containing fades.

7. A new exact-match integer 4×4 or 8×8 spatial block transform is designed to allow the precise placement of residual signals with less noise around edges (referred as "ringing" artifacts). This transform is similar to DCT design, but simplified and able to provided exactly-specified decoding.

8. Logarithmic step size control quantization or frequency-customized quantization could be selected by encoders to simplify inverse-quantization scaling or perceptual-based quantization optimization.

9. An in-loop deblocking filter is designed to prevent the blocking artifacts.

10. Context-adaptive binary arithmetic coding (CABAC) and context-adaptive variable-length coding (CAVLC) are designed as entropy coding for losslessly compressing syntax elements in the video stream knowing the probabilities of syntax elements in a given context.

11. A network abstraction layer, flexible macroblock ordering, data partitioning and redundant slices are designed as loss resilience features.

12. Support of monochrome, 4:2:0, 4:2:2, and 4:4:4 chroma subsampling.

13. Picture order count keeps the ordering of the pictures and the values of samples in the decoded pictures and allows timing information to be carried and controlled/changed separately by a system without effecting decoded picture content.

H.264 is the most advanced and efficient video compression standard. It has a very broad application range that includes all kinds of digital video compression from internet streaming applications to HDTV broadcasting. With the use of H.264, up to 50% of bit rate could be saved.

2.8 Similarity measurement

Similarity measurement is one measurement to determine the existence of watermark information in the target host. When the cross-correlation is used as similarity measurement, it can be computed as the different types of inner product of two images.

1. Linear correlation

Linear correlation is the most basic. The linear correlation between two images f and g can be described as follows:

$$r_{lc}(f,g) = \frac{1}{M \times N} \sum_x \sum_y f(x,y) g(x,y) \tag{2.50}$$

where, $M \times N$ is the size of the image. If f is the reference mark pattern and g is the extracted mark pattern from the image, r_{lc} is called similarity. One problem with linear correlation is that the detection values are highly dependant on the magnitudes of the watermark pattern extracted from the images. Therefore, for many extraction methods, the watermark will not be robust against the attacks, such as, changing the brightness of images [3].

2. Normalized correlation

This problem can be solved by normalizing the extracted mark pattern and the reference mark pattern to unit magnitude before computing the inner product between them.

$$r_{nc}(f,g) = \sum_x \sum_y \widetilde{f}(x,y) \widetilde{g}(x,y) \tag{2.51}$$

where

$$\begin{cases} \widetilde{f}(x,y) & = & \frac{f(x,y)}{\sqrt{\sum_i \sum_j f(i,j)^2}} \\ \widetilde{g}(x,y) & = & \frac{g(x,y)}{\sqrt{\sum_i \sum_j g(i,j)^2}} \end{cases} \tag{2.52}$$

We refer to Equ. (2.51) as the normalized correlation. However, normalized correlation is not robust against changes in the DC term of a work, such as, the

addition of a constant intensity to all pixels of an image [3].

3. Correlation coefficient

   The third form of cross-correlation is the correlation coefficient, which computes the cross-correlation by subtracting the means of two images before the normalized correlation.

$$r_{cc}(f,g) = r_{nc}(\widetilde{f}(x,y), \widetilde{g}(x,y)) \tag{2.53}$$

   where

$$\begin{cases} \widetilde{f}(x,y) &= f(x,y) - \overline{f}(x,y) \\ \widetilde{g}(x,y) &= g(x,y) - \overline{g}(x,y) \end{cases} \tag{2.54}$$

   where, $\overline{f}$ and $\overline{g}$ are the mean of f and g, respectively. Because the mean of an image has been subtracted, the correlation coefficient is robust against changes in the DC term of a work [3].

In this chapter, we introduced the fundamental theories and techniques used for robust digital image/video watermarking algorithms. This chapter helps understand the rest of the thesis.

Chapter 3

Literature review of image and video watermarking algorithms robust to geometric attacks and compressions

An useful digital watermarking algorithm has to be robust to many kinds of attacks. 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. Compared to image watermarking, there are more requirements specific for video watermarking, because there are a larger number of data and inherent redundancy among frames in video. A good video watermarking algorithm must be robust against video compression, frame dropping, frame swapping, geometric attacks, frame rate conversion, frame cropping, collusion attacks, noise, filtering, lighting change, histogram equalization, etc.

In this chapter, we will review and analyze some digital watermarking algorithms robust against geometric attacks and advanced video compressions.

3.1 Image and video watermarking algorithms robust to geometric attacks

Geometric attacks are generally difficult to handle for image and video watermarking. Very slight amount of geometric attacks could make the watermark detection fail due to the lack of synchronization. The existing image and video watermarking algorithms dealing with geometric attacks can be classified into four categories: (1). Rectification based, (2). Self-synchronization based, (3). Invariant domain based, and (4). Feature based.

Most of video watermarking algorithms that resist geometric attacks belong with the first two categories, which are rectification and self-synchronization based. Researchers use template, image registration or other algorithms to synchronize the watermarked video frames having undergone geometric attacks. The invariant domain based algorithms normally have higher computation complexity, which are good for still images. Feature based algorithms use some silent features as the reference coordinates for the watermark embedding and detection. These algorithms are used mostly for image watermarking algorithms robust against RST attacks.

3.1.1 Rectification based algorithms

Rectification based algorithms detect the geometric parameters by using template, exhaustive search, image registration or other algorithms. Then, it rectifies the attacked

signals to the original format. The watermark detection will be processed after rectification. According to geometric parameter detection algorithms, we classify the rectification based algorithms into four groups: (1). synchronization pattern based algorithms, (2). functional search based algorithms, (3). video registration based algorithms, and (4). digital object based algorithms.

3.1.1.1 Synchronization pattern based algorithms

Synchronization pattern based algorithms embed a template or a synchronization information to the host signal besides an informed watermark pattern. The template contains no information itself, but is used to detect transformations undergone by the image or video.

The approaches for RST invariant image watermarking [41][42][43][44][45][46] are quite similar to one another. Therefore, we will only introduce the approach proposed by [45] as an example. For details, please refer to the survey paper on RST invariant image watermarking algorithms [47].

Pereira and Pun [45] use approximately 14 points along two lines that go through the origin in the DFT domain at two random angles with radii varying between two random values. A linear transformation of an image will produce an inverse linear transformation in the DFT domain. Moreover, with a linear transformation, a line going through the origin will be transformed into a corresponding line going through the origin [45]. Therefore, the transformations could be detected through the relationship of two lines. Once the template is detected, these transformations are inverted and the spread spectrum signal is retrieved.

Since the traditional template based watermarking algorithms are easy to be attacked, some researchers came up with the new idea that the watermark bears with not

only the copyright information but also the geometrical information about the original image. The watermark does not concentrate strong energy into several points so that it is hard to be recognized by the attackers. Voloshynovshiy et al. [48] and [49] presented an efficient algorithm for the watermark estimation and recovering from global or local geometrical distortions. The estimation of the affine transform parameters is formulated as a robust penalized Maximum Likelihood (ML) problem, which is suitable for the local level as well as for global distortions. The watermark is periodic with blocks. When no geometrical transform was applied, the message is decoded from the extracted watermark directly. If some geometrical transform was applied, based on the local ACF (autocorrelation function) or magnitude spectrums, or by exploiting the reference watermark information at the block level, the geometrical distortion can be determined, then the retrieved watermark can processed and re-synchronized and the message can be decoded. In this way, the watermark acts as the roles of both the template and the copyright information bearer.

Synchronization pattern based video watermarking algorithms [9][50][51][52] embed templates to different domains or objects to identify the geometric parameters or indicate the watermark locations. Geometric attacks for video signals include not only image based RST attacks, but also spatial or temporal shift or cropping, frame removal attack, aspect ratio change based on the individual frame and the group of pictures or transformation of video signals.

- Deguillaume et al. [9] presented a video watermarking algorithm on 3D DFT domain of chunks of video scene. A video consists of a series of images at consecutive points of time. Therefore, the video could be considered as three-dimensional (3D) signals, in which, two-dimension (2D) refers to the space within each image and another dimension is the time [9][53]. The frequency spectrum of the 3D DFT

composes of the spatial and temporal frequency response at the same time. The 2D spatial frequency is to measure how fast the image intensity or color changes in each video frame. Higher frequencies in the time dimension refer to the fast change in temporal domain among all the frames in one GOP.

The 3D 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 one GOP is shown as follows [9]:

$$F(u, v, \gamma) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \sum_{t=0}^{T-1} f(x, y, t) e^{-j2\pi(\frac{ux}{M} + \frac{yv}{N} + \frac{t\gamma}{T})} \tag{3.1}$$

and the corresponding inverse 3D DFT is defined as follows:

$$f(x, y, t) = \frac{1}{MNT} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \sum_{\gamma=0}^{T-1} F(u, v, \gamma) e^{j2\pi(\frac{ux}{M} + \frac{yv}{N} + \frac{t\gamma}{T})} \tag{3.2}$$

The Discrete Fourier transform $F(u, v, \gamma)$ also can be expressed alternately using the exponential form as:

$$F(u, v, \gamma) = |F(u, v, \gamma)| e^{j\phi(u,v,\gamma)} \tag{3.3}$$

$$|F(u, v, \gamma)|^2 = Re^2(F(u, v, \gamma)) + Im^2(F(u, v, \gamma)) \tag{3.4}$$

$$\phi(u, v, \gamma) = tan^{-1}\Big[\frac{Im(F(u, v, \gamma))}{Re(F(u, v, \gamma))}\Big] \tag{3.5}$$

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

Fig. 3.1 gives an example of three consecutive frames and the magnitude of the 3D frequency response, respectively. Three frequency magnitudes illustrate the symmetry property of the Fourier transform. All frames are symmetry according to the DC component in the middle frame. Only this frame contains the DC component, which includes the DC component both in spatial domain and temporal domain at the same time.

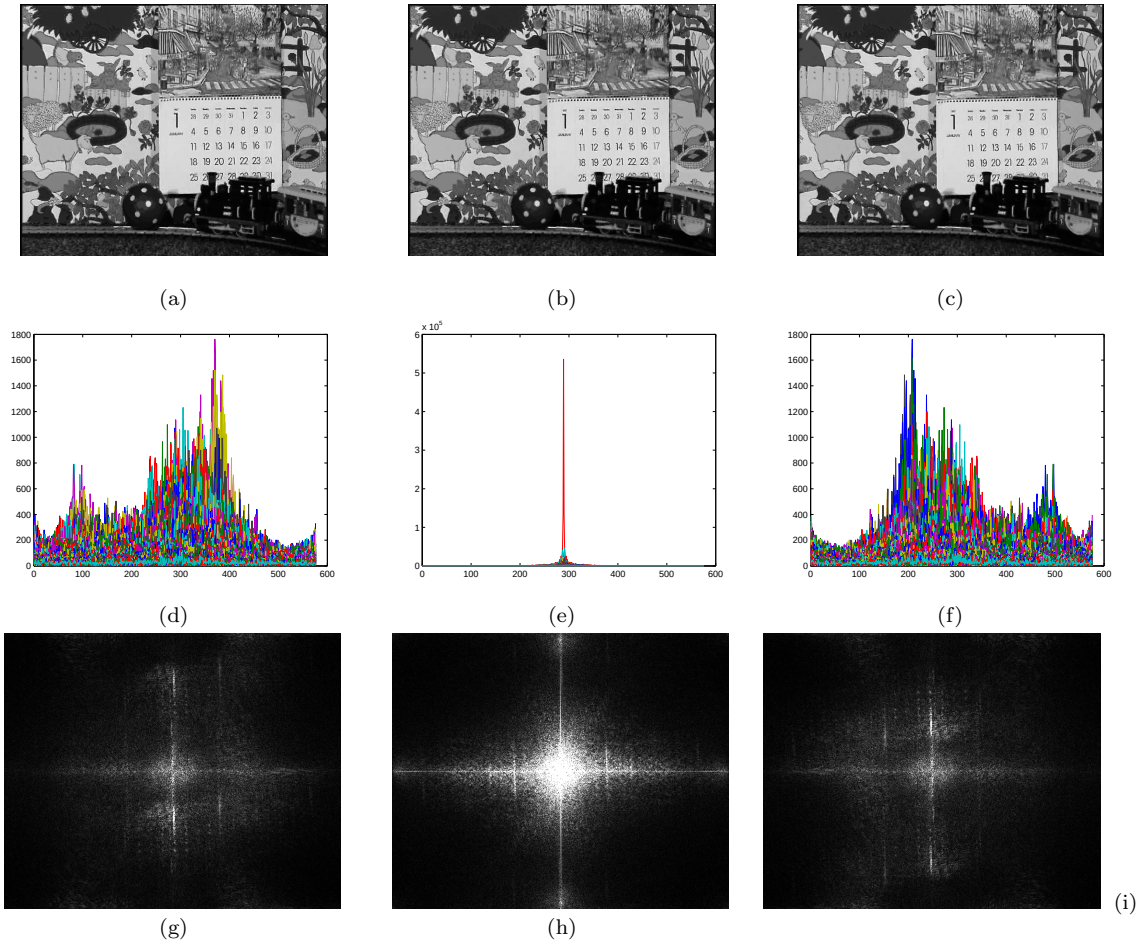


Figure 3.1: 3D DFT. (a), (b), (c) are three consecutive frames. (d), (e), (f) are 2-D plots of the corresponding magnitude of 3D DFT. (g), (h) (i) are image displays of the corresponding magnitude of 3D DFT.

Deguillaume et al. [9] embedded a watermark and a template in the 3D DFT magnitude of video chunks. A spread spectrum based watermark embedding method is adopted to provide robustness to noise or partial cancelation. The template is performed in the log-log-log map of the 3D DFT magnitude and is used to detect and invert frame-rate changes, aspect-ratio modification and rescaling of frames. This algorithm has the good performance on MPEG-2 compression combined with other attacks, such as, aspect-ratio changes, frame rate changes. The approach is more robust to averaging attacks than frame-by-frame approach and has larger bandwidth to hide data. However, high computational cost of the 3D DFT, 3D log-log-log mapping and 3D log-polar-log mapping has to be considered.

- The algorithms in [50][51] worked on raw video for geometric invariant digital watermarking. They embed a watermark in the luminance values of video frames for low computational cost and a synchronization pattern for geometric rectification.

  Zhao et al. [50] re-implemented and improved Haitsma et al.'s algorithm [54] by using an amplitude-limiting filter and a whitening filter to enhance the probability of watermark detection. Their algorithm focused on camera capturing attacks, which include filtering, shifting, scaling, rotation, shearing, changes in perspective, etc. They embed the watermark signals to the mean luminance value of successive video frames. The inherent properties of the mean luminance value are very insensitive to spatial geometrical operations. Therefore, the algorithm is inherently robust against all geometrical distortions. In Haitsma et al.'s algorithm, watermark detection is performed by correlating the watermark sequence with the extracted mean luminance values of a sequence of frames. However, if

the individual frames are randomly removed or the mean luminance values are temporally low-pass filtered, watermark detection will be failed [50]. Zhao et al. embed an informed watermark and a synchronization watermark to resist frame removal attacks. Two watermarks are orthogonal to each other and different in length. With cross-correlation between the synchronization watermark and the watermarked frame having undergone geometric attacks, the peaks indicate the location of watermark bits. The watermark can be detected successfully with correct locations. However, the algorithm depends on the luminance of each video frame. The luminance of each GOP could be different in a video clip. The false alarm could probably be high here. In other words, this algorithm may be defeated by gray level change attack.

Lancini et al. [51] also inserted a synchronization pattern and a watermark pattern to the uncompressed host signal. The synchronization pattern is used to extract the video cropping or resizing information. In order to get a good invisible watermarked video, three masking functions - luminance mask, texture mask and temporal mask are implemented prior to the watermark embedding. These functions found a lower sensitive area to noise with very high or very low brightness, highly texture and slower movement for watermark embedding. The watermark insertion is based on the spread spectrum techniques. The watermark extraction is measured on the computation of the autocorrelation of the frame and the correlation between the frame and the synchronization pattern.

- Shal et al. [52] proposed a frequency domain watermarking algorithm. In order to be robust against geometric attacks, a spread spectrum based synchronization template W_t is embedded in the frequency domain, rather than in the spatial

domain as the traditional algorithms. First, an estimated tiling template W'_t is detected by a Wiener filter. Then, the manipulations on video as scaling, aspect ratio change can be computed by using auto-correlation of W'_t. For spatial shifting or cropping, each frame can be relocated by the cross correlation function between W'_t and the original template W_t. The algorithm can be used both in image watermarking and video watermarking. As a video watermarking approach, the author used the same idea into each frame of a video sequence. This probably brings error propagation for B or P frames in a video sequence.

3.1.1.2 Functional search based algorithms

Dainaka et al. [55] proposed an algorithm based on the color planes of image or video frames. They use special functional search for rotation, scaling, translation and clipping. The same or reverse watermark pattern is embedded into many rectangular blocks of two different color planes, between which the covariance is detected. If the covariance is not zero, no matter positive or negative, one bit watermark can be detected. Two independent PN sequences are embedded to another rectangular areas called belts between blocks to be used for synchronization after clipping. Rotation and scaling search is combinatorial by so called "number of flipping" (NF) search. NF are the number of pixels which have the different signs with the pixels right and underneath. Therefore, NF search resulting in the same sign within each area indicates the correct synchronization. If it has the random positive or negative signs, the angle or size is not correctly restored. Clipping and translation are restored by horizontal and vertical NF search. Although functional search is not exactly exhaustive search, it still possibly increases the false alarm probability.

Echizen et al. [56] improved Dainaka's algorithm by using human visual system.

Their basic idea is to create the constant color changes by controlling watermarking strength in terms of color difference in $L \times U \times V$ space. The watermarking strength is replaced with the product of the average watermarking strength and the relative acceptability of the watermark at each pixel. The detection ratios are largely improved with the proposed optimal embedding algorithm.

A Radon transform based digital image watermarking algorithm has been proposed in [32]. In the algorithm, two generalized Radon transforms, Radial Integration Transform (RIT) and Circular Integration Transform (CIT), as introduced in Section 2.3, are used to extract some characteristic values, based on which the corresponding geometric transformation parameters can be calculated to re-synchronize the transformed watermarked image.

The watermark is embedded into the spatial domain. Once the geometrically distorted watermarked image is transformed back to its original shape in terms of position, orientation and size, the detection of the watermark can be very straightforward.

Based on the properties of the RIT and CIT, the geometric transformation parameters can be calculated. As shown in Equ. (2.21) and Equ. (2.22), the computation origin $f(x_0, y_0)$ of the RIT and CIT is very important for the successful detection of the geometric transformation parameters. The Harris corner detection algorithm [38], introduced in Section 2.6, is used to find the corner points. The one, which is the most robust to possible attacks, is used as the computational origin for the RIT and CIT.

3.1.1.3 Video registration based algorithms

Digital rights management (DRM) systems adopt forensic watermarking to protect and enforce copyright control with digital media content such as audio, video or still images [17]. In digital cinema application, forensic watermark is used to determine the

copyright owner, where and when the theft occurs and to prevent such illegal activities. Normally, the original video is available for watermark detection. Most forensic watermarking algorithms use video registration algorithms [18][19][20] for watermark detection.

- Cheng et al. [18] proposed a Spatial, Temporal and Histogram (STH) video registration algorithm for digital watermarking detection of the camera captured videos, which have significant misalignments with the reference videos. Spatial misalignment could be the results of warping, cropping, resizing, rotation, etc., as one image could have undergone a 8-parameter plane perspective transformation. Temporal misalignment is caused by frame rate conversion, frame dropping or frame duplication. Reference and candidate videos could have different color histogram because of compression, filtering or gamma change. Cheng's STH video registration algorithm could recover three mixed misalignments in one dynamic programming. They model temporal misalignment by using 2-frame integration model as most captured frames are a linear combination of two consecutive displayed frames. Histogram transformation is modeled by a table look-up, in which one of 256 parameters in each candidate frame could map gray levels in a reference frame. Spatial misalignment is modeled as 8-parameter plane perspective [57]. The minimum accumulated mean square error (MSE) is optimized between the reference frame and the candidate frame by dynamic programming. If MSE is lower than a threshold the iteration will stop. This algorithm is robust against most cinema capture distortion. However, the threshold decides the iteration times. If the threshold is too high, it will fail to detect the watermark. If the threshold is too low, the computation complex will be too high. It is a trade off between them.

- Nguyen et al. [19] proposed a so called IRISA video registration algorithm. They model the spatial misalignment by using a semi-automatic algorithm on the feature points, extracted using the Harris detector, introduced in Section 2.6. The spatial transformation is computed through the registration of the set of feature points between original and candidate images. The spatial luminance variation is caused by the display condition, such as, lens, the screen and ambient condition, which is modeled with non-uniformity function depending on geometric transformations. The temporal luminance variation results from automatic gain control (AGC), and the luminance function is separated in time and space. As a result, the estimate of the watermarked image can be modeled with both spatial non-uniformity function and the temporal luminance function.

- Delannay et al. [20] proposed a key frame based temporal registration algorithm. In this approach, the key frame is the histogram of its luminance component, which is significantly different from the luminance histogram of the previous frame. The temporal alignment is matched between two key frames of reference and candidate sequences by adopting Beta's computation and best matching algorithm. Assuming the orders of frames are the same among original and candidate sequences, the temporal registration is formulated as a minimization problem of the combination or error model and the penalization of large change between the reference and the original frame. The algorithm could give a good results for synchronized sequence with the adaptive and pseudo-global threshold.

3.1.1.4 Digital object based algorithms

Video object is an important concept for MPEG-4 compression standard. Video objects can be extracted, distorted and duplicated for illegal use. Several video watermark-

ing algorithms work on the geometrical invariant watermarking on MPEG-4 objects [58][59][60].

- Lu et al. [58] proposed their video watermarking algorithm based on eigenvectors of video objects for synchronization of rotation and flipping. The eigenvectors of a video object are related to mean vector and the covariance matrix of a given video object, and are calculated to determine its major and minor orientation tendencies. The major eigenvector is aligned with the positive x-direction and the minor eigenvector is with vertical (up or down) direction. In the detection process, the video object will be adjusted such that the eigenvectors can be in the accordance with the pre-determined directions. As a result, the synchronization of rotation and flipping is obtained. The watermark is embedded into the DCT coefficients with the middle frequencies during MPEG-4 compression. A mean filtering operation is proposed in order to blindly extract the signal for a suspect area in the watermark detection process. The detection results show the strong robustness against flipping, blurring, and histogram equalization, but it is not very robust to rotation, additive noise, rescaling and MPEG compression.

- Hsu et al.'s algorithm [59] derived the eigenvectors of the video objects from 2-D and 3-D principal component analysis (PCA) of the objects. The direction of each eigenvector derived from 2D PCA of the shape will change after the video object is distorted by the geometric transformation, which make it difficult to synchronize the objects orientation. Therefore, they proposed the principal component analysis to incorporate one more dimension of image intensity. 3-D eigenvectors derived from 3-D PCA of the object are employed to calibrate the 2-D coordinate system. During watermark embedding, they use the intersection

of the eigenvectors and the object boundary to decompose the object into non-overlapped regions. A triangle from each decomposed region is extracted and a triangular watermark with one-bit information is embedded to each triangle independently. The tessellation triangle watermark embedding algorithm was first proposed by P. Bas et al. [61]. Bas et al. generated the triangles by the feature points detected by Harris detectors. However, the feature points are not stable under attacks, such as, filtering, noise and compression. 2-D eigenvectors are more stable than feature points under attacks.

- Boulgouris et al. [60] also worked on MPEG-4 video objects. Both a synchronization information and a copyright watermark pattern are inserted in the luminance of video frames. An independent object center is calculated as the mean of the coordinates of all non-zero pixels of the binary alpha mask. The synchronization information is embedded in a meander path around the center. The position of the synchronization information is independent of the spatial location of the object. During the detection, the synchronization signal is searched with the maximum correlation metric with the candidate center as the starting point of the original embedding . This algorithm is robust against frame cropping, video format conversion from MPEG-4 to MPEG-2. However, this algorithm uses exhaustive search for synchronization signal location in a small area. False positive probability could be high.

3.1.2 Self-synchronization based algorithms

Because video has much more information than image, complex embedding and detection algorithm for video will have higher computational cost. Self-synchronization

based algorithms do not depend on the synchronization pattern or template to find out the geometric parameters to invert the signal to the original format and do not transform the video frames to the geometric invariant domain which is usually complex frequency domain or polar mapping. Therefore, self-synchronization based algorithms usually have less computational complexity, and are more suitable for video watermarking algorithm robust to geometric attacks.

3.1.2.1 Compressed domain based self-synchronization algorithms

Some video watermarking algorithms [8][62][63] work on compressed video sequence and embed watermark to DCT blocks, frames of GOPs, or motion vectors. These algorithms avoid the compression/decompression procedures and are able to conduct real-time video watermarking as the benefit.

- Langelaar et al. [8] proposed the differential energy watermarking (DEW) algorithm for JPEG/MPEG streams. This algorithm was initially designed for still images and has been extended to video by watermarking the I-frames of a MPEG stream. It works directly on DCT blocks and avoids the need for JPEG/MPEG decoding. Therefore, it is suitable for real time watermarking. The DEW algorithm discards high frequency DCT coefficients in the compressed data stream, and embeds label bits according to the energy difference between the top half block and the bottom half block of DCT coefficients. The minimal cutoff is considered with balance of more robust watermark and image degradation. They embed a single information bit into more 8×8 blocks instead of one to be robust to watermark removal attacks. According to the experimental results, this algorithm is robust against JPEG/MPEG, recoding, and Stirmark attacks. However, the bit errors for Strimark attacks are averaged at 20% for tested images.

- Wang et al. [62] proposed a blind video watermarking algorithm for MPEG-2 videos. In order to be robust against cropping, the watermark sequence is redundantly embedded in each individual rows of I-frame during MPEG-2 compression. Due to the separable property of DCT, 1D DCT along the horizontal direction can be obtained from 2D DCT block. Each 1×8 1-D DCT works as the watermark embedding location. In this way, no matter which rows or how many rows are removed by horizontal cropping, the watermark still could be detected. To be robust against downscaling, the watermark is embedded into the low frequency area of full DCT domain since the spatial downscaling of one frame has roughly equivalent to the truncation of high-frequency band in full DCT domain. The conversion from block DCT to full DCT is needed as the cost. To deal with frame dropping, the algorithm divides the video into N scenes, then, redundantly embed the same group of bits in each frame within the same scene. This algorithm can self-recognize the watermark structure during the detection. However, it is limited to one compression standard. It may not work under format conversion.

- El'arbi et al. [63] proposed an algorithm resistant to geometric transformations based on motion activities and wavelet network training. A motion activity matrix obtained in the video pre-processing saves the coefficient position with the information in the frame where it stops moving and its motion activity. During watermark embedding, motion activity matrix allocates the embedding location with the maximum motion intensity for the first intra-frame. Motion compensation decides the corresponding new positions of the watermark. If any coefficient stops moving, other new coefficients will be selected for watermark embedding. In this way, the watermark location is independent from geometric distortion, but

related with motion activity and motion compensation. This algorithm is robust against aspect ratio change, rotation, resizing and cropping.

3.1.2.2 Raw video based self-synchronization algorithms

Other video watermarking algorithms [12][64][65][66] worked on the raw data. These algorithms do not limit to any compression standard.

- Kalker et al. [12] proposed a video watermarking system called Just Another Watermarking System (JAWS) in Philips Research Laboratory. A watermark in JAWS is simply additive noise and it uses floating point values with benefit that a normal probability distribution has the largest entropy under the condition that the average energy of the watermark per pixel is equal to 1. A Laplacian high-pass filter is applied to the watermark embedding processing to remove the artifacts of image regions with little activity. In order to be shift invariant, the JAWS embeds the tiling watermark in each frames. This algorithm is robust to vertical and horizontal shift. The JAWS detection operates in the digital base-band domain and simple spatial correlation. It has less computational complexity and could be real-time. However, it is not robust against rotation or scaling.

- The algorithm of Zhuang et al. [64] is based on video scene segmentation and 3D wavelet transform. They proposed a spatial-temporal embedding and blind detecting algorithm for video watermarking. The watermark is embedded into randomly selected scene after scene change detection. The 2D DWT is applied to each frame in the selected scene and the 1D DWT is applied furthermore to the selected area in the 2D DWT coefficients for blind watermark embedding and detection. They change the corresponding coefficients of the video shot adaptively

to embed the watermark . This algorithm is robust to key attack, noise attack, frame cropping, frame dropping, frame averaging, frame swapping and MPEG compression, but not geometric attacks.

- Su et al. [65] proposed a content dependent and spatially localized video watermarking algorithm with a frame-by-frame strategy. In order to prevent statistical collusion, they use similar watermark pattern for similar frames and the designed statistical correlation of watermarked frames must match that of the host video frames, shown as follows:

$$\rho(X_i, X_j) = \rho(U_i, U_j) \tag{3.6}$$

for all frame indices i, j , where X is the watermarked frame as $X = U + \alpha W$, U is the host, W is the watermark, and $\rho(A, B)$ is the correlation coefficient between A and B. Therefore, content-dependent spatially localized watermarking algorithm is proposed. The basic geometric attacks such as rotation and pixel translation can be represented by two operations: a change of sampling grid and a re-sampling by interpolation. This algorithm places the watermark into the low distortion regions by looking for the least distorted subframe by interpolation noises. This algorithm could be robust against signal enhancement, JPEG compression, cropping up to 50%, rotation with small angles and flipping. It shows a better performance than JAWS watermark. However, they did not show the experimental results on the key video processing, such as, frame averaging, frame dropping or MPEG compression.

- The algorithm of Liu et al. [66] is based on the 3-level DWT of the Y component of each frame in the target video sequence. A spread spectrum watermarking is

proposed in LL subband to be robust against frame loss. The BCH code is applied to reduce error probabilities and 3D interleaving technique is exploited to combat bursts of errors. The synchronization information is embedded to the host video to increase the robustness to temporal attacks, such as, frame dropping, frame swapping and frame inserting. The watermark pattern is modulated to be random and more secret by a random binary sequence. During watermark embedding process, some least significant bits of the selected DWT coefficients are replaced by the representative value of binary "1" or "0". This algorithm is robust to MPEG-2 compression and scaling up to 2.0. However, embedding watermark in the LL subband of DWT domain could degrade the quality of watermarked video.

3.1.3 Invariant domain based algorithms

Invariant domain based algorithms are mostly for RST invariant image watermarking. We will only briefly introduce them in this thesis. For details, please refer to the survey paper on RST invariant image watermarking algorithms [47].

3.1.3.1 Full-invariant domain based algorithms

This category includes the watermarking algorithms that embed watermark in domains that are invariant to geometric attacks. It is well known that the Fourier transform has the property of shift invariance. The Fourier-Mellin transform (FMT) has the property of rotation and scaling invariance. So once the DFT and the FMT are applied to the image, the image will be transformed to the RST invariant domain. The watermark embedded into this domain can be RST invariant. O'ruanaidh et al. [5] and Kim et al. [67] proposed two full-invariant domain based algorithms on RST image watermarking.

3.1.3.2 Semi-invariant domain and image registration based algorithms

Because of the implementation difficulties, some modified algorithms based on the FMT are proposed. Most of them use the log-polar mapping (LPM) instead of the FMT, which is not truly RST invariant. However rotation and scaling in the spatial domain result in a shift in the LPM domain, which simplifies the watermark detection dramatically. The shift in the LPM domain can be dealt with easily with image registration related techniques or the one-dimensional projection. Since these algorithms are derived from the FMT, we discuss them in this section given the fact that the LPM domain is not truly RST invariant. The image registration techniques can identify the shift in the LPM domain. Then we can re-synchronize the watermarked image to detect the watermark. Zheng et al. [29] and Liu et al. [68] use the phase correlation or phase-only matching filtering to re-synchronize the watermarked image. Also the one-dimensional projection can simplify the shift in the two dimensional LPM domain to the shift in one-dimensional projection. Exhaustive search detects the watermark since the possible positions of the watermark are confined to the one-dimensional projection. A related watermarking algorithm is described by Lin et al. [1]. We can call the LPM domain or the one-dimensional projection of the LPM domain the semi-RST invariant domain.

3.1.3.3 Image decomposition based algorithms

Another approach to a RST invariant watermarking algorithm is to decompose the image or watermark into components using a set of orthogonal or non-orthogonal base functions. These decomposed components can have some RST invariance properties. The correlation between the decomposed components of an image (kth circular harmonic function, introduce in Section 2.4) and the image will not be effected by rotation. The Pseudo-Zernike basis is a set of complete and orthogonal functions. The expansions

of the image based on the Pseudo-Zernike [69] basis have the properties of RST invariance. Kim et al. [36], Xin et al. [70] and Alghoniemy et al. [71] proposed approaches to geometric transformation invariant watermarking. In these approaches, the watermarking is based on invariant parameters extracted from the geometric moments of the image. Also, other moments, like complex moments, are used for pattern recognition [72][73]. Similar watermarking algorithms have been proposed [74][75][76][77].

3.1.3.4 Stochastic analysis based algorithms

The stochastic characteristics of the image are very important for image analysis. Local mean and local variance represents the image spatial distribution. The moments of an image implement RST invariant watermarking algorithms.

Higher order spectra are defined in terms of the higher-order moments or cumulants of a signal, and are used for identification of nonlinear, non-Gaussian random processes and deterministic signals [78].

Kim et al. [79] proposed an image watermarking algorithm that is resilient to rotation, scaling and translation (RST) by using the higher order spectra (HOS) in particular bispectrum. The translation and scaling invariance is achieved by using the phases of the integrated bispectra while rotation invariant is obtained by using the Radon transform of the image.

The moments of objects have been widely used in pattern recognition, image registration [80], and image watermarking [71][81]. Alghoniemy et al. [71] proposed geometric invariance in image watermarking based on moments and image normalization. They use geometric moments to normalize geometrically the image before watermark embedding at the encoder and before watermark detection at the decoder.

The idea of [81] is to transform geometrically the image into a standard form no

matter how the image has undergone RST attacks. The translation invariant can be achieved by using the central moments of the image, which are origin independent. The scaling normalization transforms the image into its standard form by translating the origin of the image to its centroid $(\overline{x}, \overline{y})$.

3.1.4 Feature based algorithms

The template based watermarking algorithms are trying to add a recognizable template into the host image, and this template bears with some information about the geometrical structure of the host image. Given the assumption that the template can always be retrieved, based on the distortion applied to the template, we can detect the geometrical transforms the image has undergone. Some patterns possessed by host signals can also be extracted and used as the reference templates. Since this pattern has to be recognizable, normally we use some salient features as the pattern so that we can identify this pattern even the host image is severely distorted. The location of the watermark is not linked with image coordinates, but with image features or semantics [61][82][83]. The problem of geometrical synchronization can be solved because the image features represent an invariant reference to geometrical transformations. The feature could be some feature points extracted through the corner or edge detection algorithms. The following two watermarking algorithms use the feature points as the reference coordinates of the watermark embedding location.

Bas et al. [61] proposed a geometrically invariant watermarking algorithm by using feature points and Delaunay tessellation. The important step of this algorithm is to choose Delaunay tessellation. The tessellation has the two properties: 1) if a vertex disappears, only the connected triangles are modified; 2) if the vertex is moving inside the triangle area, the tessellation is not modified.

A feature extraction algorithm called Mexican Hat wavelet scale interaction is used in the algorithm in [84]. The extracted feature points can survive a variety of attacks and can be used as reference points for both watermark embedding and detection. The normalized image of an image (object) is nearly invariant with respect to rotations. As a result, the watermark detection task can be much simplified when it is applied to the normalized image. Since the loss of information caused by the cropping will cause the inaccuracy of the image normalization, the image normalization is applied to non-overlapped image regions separately in this algorithm. The regions are centered at the extracted feature points. These points are salient feature points and the small circle regions centered at these points are unlikely to be effected by cropping which normally occurs around the brim area of the image. Also the multiple circle regions can work as redundancy to increase the possibility of the successful watermark detection. After image normalization, the geometrically transformed circular regions will have the same direction, size and orientation as the original circular regions. So the regions for watermark embedding can be located after rotation and the watermark can be detected. While this strategy can solve the problem of cropping, it introduces other problems. The origins of those circular areas are the extracted feature points. This increases the possibility of the inaccuracy since the interpolation used in rotation and scaling will cause the shift of the feature points. It is worth noting that this algorithm does not modify the moments of the image directly, it instead computes the Cartesian coordinates of those pixels to be modified based on their coordinates in the normalized domain. So the watermark is embedded into the spatial domain (or frequency domain after some orthogonal transform). In this way, the distortion caused by those unorthogonal (or deviation from orthogonality caused by computation in digital domain) transforms or coordination changes can be avoided during embedding.

Several other watermarking algorithms are feature-based. They are so-called the second-generation watermarking algorithms because the feature of an image is exploited for embedding watermark [85]. Duric et al. [86] have proposed an algorithm for recognizing geometrically distorted images and restoring their original appearances by using image feature points. Sun et al. [87] have developed an algorithm based on image feature to identify the geometrical transformation. Alghoniemy et al. [88] introduced a RST synchronization algorithm based on the wavelet decomposition of an image. Dittmann et al. [89] have designed a content-based watermarking algorithm that does not require the original image and uses self spanning pattern.

There are some limitations for the feature point detectors. A geometric transform may vary the results of the feature point detection, which will result in false feature points and failure of watermark detection. Scaling and local distortion specially effect local operators significantly. Segmentation-based feature point extraction, which uses a segmentation of the image to determine the feature points, can provide more stable points. For instance, the centroid of each region identified through segmentation may be selected as a feature point. The segmented regions are relatively invariant to image manipulations. The centroid positions are more accurate references to the positions of the watermark than the feature points when the image has undergone some geometrical transforms. For example, Gibbs Random Field (GRF) based image segmentation algorithm segments an image into spatially contiguous regions. The centroid of the regions can be selected as feature points.

3.1.5 Summary of image and video watermarking algorithms robust to geometric attacks

Most recent video watermarking algorithms belong with first two categories. The rectification based algorithms normally present more robust to geometric attacks than the self-synchronization based algorithms because these algorithms could rectify the distorted signal with the information provided by extra synchronization pattern, special function or video registration with original video. However, requiring the synchronization pattern or original video as extra cost is the drawback of the rectification based algorithms.

The invariant domain based algorithms work better for still images with high computation complexity. The feature based algorithms are mainly used for image watermarking algorithms so far because it is hard to keep the same salient feature points in a group of pictures. However, there may be some ways to use these two categories of algorithms for video watermarking, which will be our future work.

3.2 Video watermarking algorithms robust to compressions

Due to the huge size of video clips, video sequences must be compressed to low bit rates before data transmission through most channels. Video compression is being developed to get better video quality with lower bit rates. MPEG-2, MPEG-4, and H.264 are most popular video compression standards so far. Many researchers research on video watermarking algorithms robust to video compressions.

Compared with MPEG-4 and H.264, MPEG-2 has the higher bit rates and lower

video quality. MPEG-4 standard provides object-based coding for audio, video and graphics. H.264 is the standard with the highest performance in video data compression. H.264 offers twice the compression ratio of MPEG-2 and H.263 at the same video quality [40].

In the previous section, we reviewed some video watermarking algorithms robust against geometric attacks. In this section, we will review video watermarking algorithms robust to video compressions, such as, MPEG-2, MPEG-4 and H.264 compression. Some of algorithms [19][20][50][54][55][56][63] mentioned in the previous section did not deal with compression at all. Su et al. [65] claimed their work robust against JPEG compression. Some other researchers [9][12][52][62][66] proposed their algorithms robust against MPEG-2 compression with different bit rates. Other algorithms [18][51][58][59][60][64] were proposed to be robust against MPEG-4 compression. However, none of above algorithms claims to be robust to both geometric attacks and H.264 compression. We will introduce some video watermarking algorithms [90] [91][92][93][94][95][96][97] resistant to H.264 compression, but not robust to geometric attacks.

3.2.1 Video watermarking algorithms robust to MPEG-2 compression and geometric attacks

In this section, we review the video watermarking algorithms robust against both MPEG-2 and geometric attacks. The solutions of these strategies against geometric attacks have been reviewed in the previous section. We will focus on only MPEG-2 attacks here.

Several researchers [52][66] worked on raw video and use special techniques, such

as error control coding, to deal with compression. Shao et al. [52] proposed a spread spectrum based video watermarking. The watermark is embedded into each frame of a video sequence. B-frames have the highest error bits because they are suffered from motion compensation the most. In order to solve this problem, authors employ repetitive coding in inter frames to accumulate detection results along the time axis and BCH (63,30,6) for error correction with capability of 6. In this way, the watermark can be successfully detected when MPEG-2 compression is down to 200 kbps without any other attacks. With the combinational attacks of translation (in both vertical and horizontal directions) plus MPEG-2 (coded at 300 kbps), and scaling to 50% plus MPEG-2 (coded at 300 kbps), the watermark still can be detected without error. The BCH coding is a highly flexible coding algorithm. Customers are allowed to control the block length and acceptable error threshold [98]. The watermark signal is converted to binary sequence and then encoded by the BCH code. Syndrome code is applied for decoding. Liu et al. [66] also employ the BCH code to reduce error probabilities in their DWT-based video watermarking algorithm. They use 3-D interleaving to eliminate burst of errors. The byte error rate is less than 1% when MPEG-2 coding is applied to the watermarked video with a bit rate of 2.7 Mbps.

Other algorithms [9][12] work on the raw video as well. Strategies against compression are highly related to how the watermark embedding and detection work. Deguillaume et al. [9] proposed a robust 3D DFT video watermarking. The watermark is encoded as a spread spectrum signal. The template is matched in the log-log-log map of the 3D DFT magnitude. In this way, an efficient template search is provided for the watermark detection. The spread spectrum watermark signal is robust against lossy compression. Accept for the high computation complexity, the algorithm is pretty robust to geometric attacks or normal video processing combined with MPEG-2 com-

pression. Kalker et al. [12] proposed so-called JAWS watermarking system on the raw video. Tiling patterns of watermark matrix are embedded to the video signal for shift invariance. Similar to the spread spectrum, this embedding algorithm also enhances the ability of robustness to lossy compression. It makes JAWS system survive MPEG-2 compression with a bit rate of 2 Mbps.

Wang et. al [62] proposed a video watermarking algorithm on MPEG-2 video bit stream. The watermarking algorithm is inherently robust against MPEG-2 compresion. Focusing on the MPEG-2 video, they proposed a video watermarking algorithm robust against geometric attacks. They adjust quantization steps to balance the flipping threshold, the watermark extraction error and the watermarked video quality. The test results show that they get 0% average error rate on bit rate reduction from 6 Mbps to 4 Mbps and format conversion from MPEG-2 to AVI.

3.2.2 Video watermarking algorithms robust to MPEG-4 compression and geometric attacks

MPEG-4 is a more advanced compression standard than MPEG-2. It absorbed many features from MPEG-1 and MPEG-2, and added more features. The principal feature of MPEG-4 is object-oriented. Based on this feature, many researchers work on MPEG-4 objects and the related attacks. Objects are extracted from the video frames by using binary or gray-scale masks, while binary masks are usually used [60]. The watermark embedding and detection are employed in MPEG-4 sequence.

The authors of [58][59][60] proposed the video watermarking algorithms based on MPEG-4 video objects. Eigenvectors of video objects are calculated and utilized for synchronization of geometric transformation in papers [58][59]. Lu et al. [58] claim

their algorithm is robust against blurring, histogram equalization, and flipping. However, the test results for other attacks, such as, filtering, scaling, MPEG compression, noise, rotation, are not very good. Hsu et al. [59] embed watermark into the tessellation triangles and claim their proposed algorithm robust against MPEG-4, rotation, flipping, scaling, and general image processing attacks. Boulgouris et al. [60] embed synchronization information around the center of the object to find out the starting point of watermark sequence. The center of the object is calculated as the mean values of all the non-zero pixels of the binary alpha mask for object extraction and is independent of the bounding box of video object during MPEG-4 encoding. Re-synchronization is independent and robust against transcoding from MPEG-4 to MPEG-1/2.

Not like the above mentioned algorithms working on video objects in MPEG-4 sequence, other papers [18][51][64] embed the watermark in the raw video. They use error control coding (ECC) or their strong embedding and detection algorithm to provide robustness against MPEG-4 compression. Lancini et al. [51] embed the watermark pattern into the luminance component of each frame because the chrominance quality is normally decreased significantly by compression. In order to improve the signal to noise ratio after signal transmission though the noisy channel, they test two kinds of ECCs: convolutional code and turbo code during watermark detection. Convolutional codes convert m-bit data stream into n-bit codeword while m/n is the code rate. Viterbi algorithm is usually the decoding strategy for it. Turbo codes are a class of high-performance forward error correction (FEC) codes. They are concatenated codes with convolutional code and a random interleaver. A soft output Viterbi algorithm (SOVA) is used for decoding the turbo codes here. A synchronization pattern is inserted to synchronize the geometrical transformed watermarked images. The test results show that for the small block less than 192 bits, the convolutional code are better than the

turbo code. For higher video compression ratio, the turbo code is much better than the convolutional code. The bit error probability is lower than 10^{-2} when the target videos are re-compressed using MPEG-4 at a bit rate of 100 kbps after MPEG-2 compression at 1 Mbps.

Cheng et al. [18] proposed a Spatial, Temporal and Histogram (STH) video registration algorithm on the raw video. They model temporal, histogram and spatial misalignment first, then create STH video registration to minimize the matching cost and local prediction error. The detection results show this algorithm is robust against MPEG-4 until the bit rates reach 1 Mbps. Zhuang et al. [64] work on the raw video to propose a spatial-temporal algorithm based on 3D wavelet transform and scene segmentation. The 3D DWT is the combination of 2D DWT on each frame and 1D DWT across the temporal axis in the selected area. The watermark is spread spectrum embedded in the 3D DWT coefficients. This algorithm is robust against MPEG compression with an average PSNR of 20 dB.

3.2.3 Video watermarking algorithms robust to H.264 compression

To the best of our knowledge, there is no video watermarking algorithm that is robust against both geometric attacks and H.264 compression. In this section, we review some video watermarking algorithms [90][91][92][93][94][95][96][97] that only focus on H.264 compression. All these algorithms embed the watermark in the H.264 stream, therefore, they are inherently robust against H.264 compression, which is so called native H.264 video watermarking [95]. Based on the features of H.264 compression, researchers use different H.264 compression stages as their watermark embedding locations. Most of

them embed the watermark into the DCT coefficients, either 4×4 or 16×16 macroblocks. However, Mohaghegh et al. [96] proposed their video watermarking algorithm on motion vectors of B/P macroblocks and Mohammad et al. [97] embedded the watermark on the residual DC values of the 4×4 intra-prediction blocks.

Lu et. al [90] proposed a blind video watermarking algorithm for H.264. The watermark embedding or detection process is integrated in H.264 encoder and decoder. They embed the watermark into the quantized AC coefficients of 4×4 DCT blocks because of the similarity of the entire 16×16 macroblock between neighbors. Block polarity is "0" if the quantized DC coefficient is "0", or defined as "1" is the quantized DC coefficient is not "0". "1" block polarity indicates the candidate for watermark embedding. Block index modulation modifies the least significant bit (LSB) of the activity index to force the activity to be quantized into a specific region. Dead zone evacuation is used to enhance the robustness. This algorithm produces a good quality of watermarked video. The watermark detection is conducted without need for original video. It is also robust against re-encoding and Gaussian noise attacks.

Qiu et al. [91] proposed a hybrid video watermarking algorithm by embedding a robust watermark in the DCT coefficients of I-frames for copyright protection and a fragile watermark in motion vectors of P-frames for authentication during the H.264 encoding process. They insert the robust watermark bits into the quantized AC coefficients of luminance blocks within I-frames. The watermark signal is set to be strong enough for quantization to re-compression. In order to be robust against transcoding, they select to modify the coefficients in diagonal positions during watermark embedding. For the fragile watermarking, they insert the watermark by modifying one component of selected Motion Vectors (MV). The optimal MVs are selected with minimal Lagrangian cost function, which is the combination of distortion cost and the number of bits to

encode the MV prediction error. The test results show that this algorithm can resist the re-compression with fixed quantization parameters and the fragile watermark is easy to be removed. The benefit of the small block size of 4×4 integer DCT transform is to reduce the ringing artifacts, which are normally "echo" or "ghost" near edges. However, the embedded watermark are more sensitive to other attacks, such as, noise addition, filtering, and sharpening [99]. Noorkami et al. [92] also embed the watermark in the AC coefficients of DCT blocks of I-frames during H.264 encoding. They use private key and public key to decide the randomness of the selected coefficient for watermark embedding. These keys are determined by the relative differences of the DC coefficients, which have to be sent to the watermark detection side as the side information. This algorithm is robust to H.264 re-encoding. However, it is not robust against any other simple signal processing operations.

Zhang et al. [93] proposed a robust video watermarking algorithm to embed the pre-processed grayscale watermark pattern into video in the compressed domain. In order to effectively increase the robustness to the high compression ratio of H.264, the watermark sequence is pre-processed prior to the watermark embedding. The binary watermark sequence is obtained from a grayscale pattern after few transformation steps, which include the uniquely used 4×4 integer DCT for H.264 and zigzaging, normalization and frequency masking to discard the high frequency coefficients and keep the four normalized coefficients with the largest values, transformation to keep all the values positive and narrow down the dynamic range between of every two adjacent coefficients, which is caused by the masking, and level reduction to binary sequence. During the watermark embedding process, only one certain middle frequency in the diagonal positions in a 4×4 DCT block is substituted by the product of the spread-spectrum watermark and the positive gain factors. The local gain factor increases for relatively

dark and textured regions and decreases for bright and smooth regions. H.264/AVC standard supports special tree-structured motion compensation with seven inter prediction modes and two intra prediction modes for each macroblock B [40]. They use Lagrangian Optimization technique [40] to choose the best embedding mode with minimum distortion and consumed bits for encoding the current mode.

The proposed algorithm showed the robustness to bit-rate reduction, contrast enhancement and Gaussian noise addition with all watermarked video streams at 768 kbps. Hsieh et al. [94] also proposed a low complexity watermarking algorithm in H.264 compressed domain. In order to reduce the complexity, they choose 4×4 DCT blocks of I-frame for watermark embedding. Except the DC coefficient, the remaining 15 AC coefficients are modified for watermark embedding. They define a coding block embeddable if the watermark embedding function of the coefficients is greater than a threshold. A bit of 1 or 0 is embedded according to the weighting functions of the embeddable block. It is reported to be robust against re-encoding with bit-rate reduced to half of the original.

Sakazawa et al. [95] also proposed a H.264 native video watermarking algorithm to embed the watermark into H.264 stream. They embed watermark based on spatiotemporal DCT Coefficients Alteration. Their main contribution is the analysis of noise propagation and drift compensation. H.264 employs an inter-block prediction scheme to decrease the spatial redundancy. In order to avoid the spatial noise propagation, Sakazawa et al. do not embed the watermark to both two adjacent predicted blocks in the same frame. To limit the accumulated drift among frames, the drift compensation is applied to control the errors by subtracting the quantized watermark value from the original DCT coefficients of the predicted blocks. The algorithm is claimed to be robust against Gaussian filtering, median filtering, sharpening, recompression, etc.

DCT domain based watermarking algorithms, which insert watermark into DCT coefficients, can be robust against some attacks, such as, bit rate conversion transcoding and filtering. However, it is restrained by its low payload in low bit-rate applications, in which very few nonzero DCT coefficients are available for watermark embedding. Researchers start to look for the new locations to carry more watermark payload. Since even the video compressed at very low bit-rate has many nonzero Motion Vectors (MV), the watermark embedding capacity can be much higher than that in the DCT domain. Mohaghegh et al. [96] embedded watermark into the motion vectors of B/P macroblocks in H.264 video sequence because common non-geometrical distortions like digital filtering, color correction and noise do not have any influences on motion vectors [65]. In order to keep the fidelity of the watermarked video, they modify the motion vectors in the direction of either horizontal or vertical movements. The adaptive threshold for each target motion vector (MV) is updated according to the previous threshold and MV magnitude. The embedding algorithm minimizes the affect on the visual quality of the watermarked video, keeps the fast decoding speed and exploits the information of inter frames efficiently.

Since DC coefficients have the least compression ratio, Mohammad et al. [97] proposed a novel algorithm of embedding watermark in the residual DC values of the 4×4 intra-prediction blocks during H.264 encoding process. They modify the Lease Significant Bit (LSB) of DC values of not only I-frames, but also P- and B- frames to produce higher payload capacity. They simulate the rate control function of H.264 to adjust quantization parameters to change dynamically the bit rate and DC residual values as well. They focus on bit rate change and fidelity of watermarked video, but not other attacks.

3.2.4 Summary of video watermarking algorithms robust to compressions

Video watermarking algorithms reviewed in this chapter could be robust against geometric attacks and MPEG-2 or MPEG-4 compression in different levels. The video watermarking algorithms could be inherently robust against video compression if the watermark embedding and detection are carried out during compression. Some video watermarking algorithms work on the raw video and utilize error control coding to improve robustness to compression. To the best of our knowledge, none of the reviewed paper claims to be robust against both H.264 and geometric attacks at the same time.

Chapter 4

A proposed video watermarking algorithm based on log-polar mapping and phase-only filtering method

In this chapter, we present an efficient video watermarking algorithm (LPMPOF thereafter) based on the log-polar mapping and the phase-only filtering method. This algorithm is extended from our image rectification algorithm [100]. We divide the raw video source to Groups of Pictures (GOP) with fixed length. The Fourier transform is applied to the I-frame of each GOP. Then, we apply approximate log-polar mapping (LPM) to the magnitude of the Fourier spectrum. In the LPM domain, rotation or scaling transformation in the spatial domain results in vertical or horizontal shift in the log-polar domain. Translation has no affect in this domain. Therefore, LPM domain is our watermark embedding domain, which is illustrated in the work flow diagram in

Fig. 4.1. We use image registration method with a small template cut from the original frame to find the watermark locations. The phase-only filtering method is applied for the template matching, as shown in Fig. 4.1. Then, we re-synchronize the distorted frame during watermark detection process. This method is very robust to RST attacks and MPEG-2 compression.

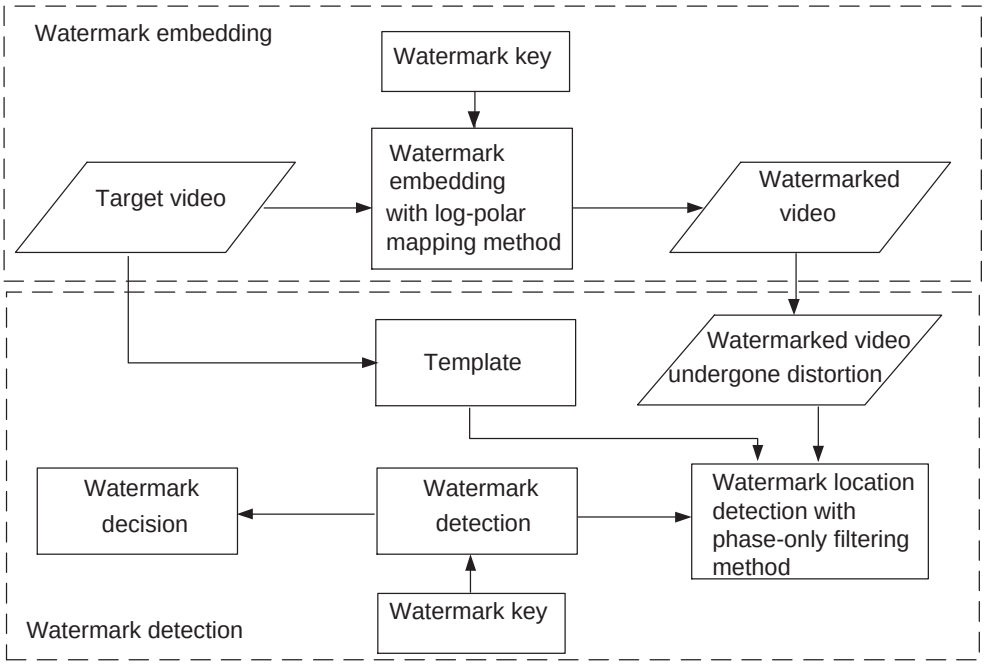


Figure 4.1: Work flow chart of LPMPOF algorithm.

4.1 Log-polar mapping method

As mentioned in Chapter 2, Fourier-Mellin transforming method, which includes 2D discrete Fourier transform, log-polar mapping on Fourier magnitude, and Fourier-Mellin transform on log-polar domain, can build a domain, which is truly invariant to Rotation, Scaling and Translation (RST) attacks. However, there is implementation difficulty for inverse log-polar mapping which is needed for inverse Fourier-Mellin transforming

method. Therefore, we have to find an alternate way, log-polar mapping method, to solve the problem. This method uses only first two steps, 2D discrete Fourier transform and log-polar mapping on Fourier magnitude. The log-polar mapping domain is a semi-RST-invariant domain, in which rotation and scaling transformations in spatial domain turn to vertical and horizontal shift, and translation in spatial domain has no affect.

However, the real LPM and ILPM still have the unacceptable computation imprecision. In order to solve this problem, we use the approximate ILPM and embed the watermark in DFT domain. According to bilinear interpolation, if we want to change the value of one point in LPM magnitude spectrum for embedding watermark data, we only need to find the corresponding four points in Cartesian magnitude spectrum and change their values accordingly, as shown in Equ. (4.1) and Fig. 4.2.

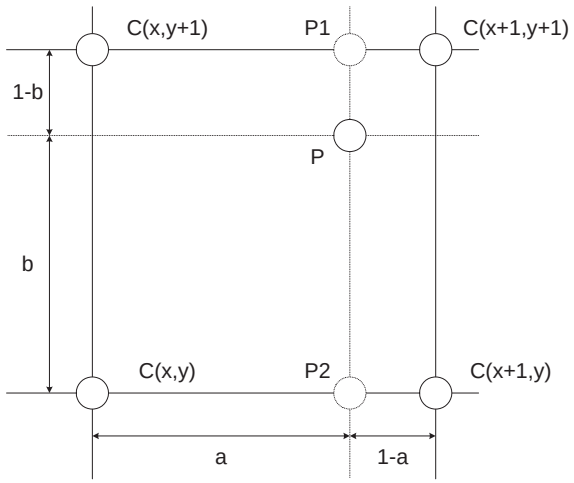


Figure 4.2: Bilinear interpolation.

$$\begin{aligned} P(\rho,\theta) &= C(x,y)\cdot(1-a)\cdot(1-b) \\ &+ C(x,y+1)\cdot(1-a)\cdot b \end{aligned}$$

$$+ \quad C(x+1,y)\cdot a\cdot(1-b)$$
$$+ \quad C(x+1,y+1)\cdot a\cdot b \tag{4.1}$$

where $C(x,y)$, $C(x,y+1)$, $C(x+1,y)$, and $C(x+1,y+1)$ are four points in Cartesian coordinate, $P(\rho,\theta)$ (P in Fig. 4.2) is the corresponding point inside the square specified by the four points, and a and b are respectively the x-axis and y-axis coordinate difference between point P and point $C(x,y)$.

According to interpolation theorem, higher order of interpolation, such as bicubic interpolation, can provide better precision. However, bicubic interpolation needs 16 points to interpolate one point. Embedding 16 watermark data into the Fourier magnitude will increase computation comparing with embedding 4 watermark data and will degrade the quality of watermarked video much more. Therefore, we choose to use bilinear interpolation.

4.2 Matching template

In the log-polar mapping domain, rotation and scaling transformation in the spatial domain turn to vertical and horizontal shift. In order to find the scaling and rotation parameters of a frame having undergone the RST attacks, a template in the LPM domain and a template matching method are needed. A small block from the selected position of the LPM domain as a matching template is cut during the watermark embedding process. When extracting the watermark, we calculate the cross-correlation between this matching template and the LPM spectrum of the Fourier transform of the watermarked frame to get the translational shifts of the template in the LPM domain. Therefore, we can obtain the rotation and scaling parameters according to the

relationship between the spatial domain and LPM domain.

Fig. 4.3 shows where we cut the matching template. This template is necessary for our rectification scheme to obtain the rotation and scaling parameters with our template matching method. Normally, we cut a template from low and middle frequency of LPM domain of a truncated I-frame because the high frequencies will be significant effected by the compression and watermark embedding. The reason why we use the truncated frames is explained in step 3 of the watermark embedding process. The template for each I-frame will be sent to the receiver side with the watermark sequence key.

In our implementation, the size of the template is 64×64 with 8 bpp. So the file size of the template data is 4096 bytes. In practice, it can be compressed to minimize its data file size without losing computation accuracy in the phase-only filtering method. In this thesis, we simply use the JPEG compression method to compress the template to 1 KB. All following experiments are carried out under this compression ratio.

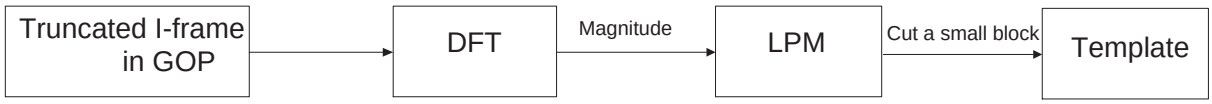


Figure 4.3: Matching templates.

4.3 Filters design

The key technique is to match the template and the watermarked frame having undergone RST attacks. In this section, we will discuss this technique.

4.3.1 Traditional filters

For a template g and an image f, where g is smaller than f, the template could be matched in the image by using the two dimensional cross-correlation function:

$$r(u,v) = \sum_x \sum_y f(x-u, y-v) g(x,y) \tag{4.2}$$

If the template matches the image exactly at a translation (i,j), the cross-correlation will have its peak at $r(i,j)$. The major disadvantage of the cross-correlation method is its computational intensity. According to the correlation theorem, the Fourier transform of the correlation of the two images is the product of the Fourier transform of the one image and the complex conjugate of Fourier transform of the other.

$$r = \mathfrak{F}^{-1}[F(w_\rho, w_\theta) \cdot G^*(w_\rho, w_\theta)] \tag{4.3}$$

where

$$\begin{cases} F(w_\rho, w_\theta) &= \mathfrak{F}(f(\rho,\theta)) \\ &= A_F(w_\rho, w_\theta) e^{-j\Phi_F(w_\rho, w_\theta)} \\ G(w_\rho, w_\theta) &= \mathfrak{F}(g(\rho,\theta)) \\ &= A_G(w_\rho, w_\theta) e^{-j\Phi_G(w_\rho, w_\theta)} \end{cases} \tag{4.4}$$

and $\mathfrak{F}$ and $\mathfrak{F}^{-1}$ are DFT and inverse DFT, respectively, $*$ is the complex conjugate, and $G(w_\rho, w_\theta)$ is called a matching filter.

The fast Fourier transform based methods are fast and efficient. The following defines five types of traditional filters [35]:

1. Classical matched filter, which is the Fourier transform of the template g, with full phase and amplitude.

$$G(\omega_\rho, \omega_\theta) = A_G(\omega_\rho, \omega_\theta) e^{-j\Phi_G(\omega_\rho, \omega_\theta)} \tag{4.5}$$

2. Amplitude-only filter, which is defined as the amplitude spectrum of the Fourier transform of the template g.

$$G_A(\omega_\rho, \omega_\theta) = A_G(\omega_\rho, \omega_\theta) \tag{4.6}$$

3. Inverse filter, which is the division between the phase spectrum and the amplitude spectrum of the template g.

$$G_I(\omega_\rho, \omega_\theta) = \frac{e^{-j\Phi_G(\omega_\rho, \omega_\theta)}}{A_G(\omega_\rho, \omega_\theta)} \tag{4.7}$$

4. Phase-only filter, which contains only the full phase spectrum of the template g.

$$G_\Phi(\omega_\rho, \omega_\theta) = e^{-j\Phi_G(\omega_\rho, \omega_\theta)} \tag{4.8}$$

5. Binary phase-only filter, which contains 1 or -1 according to the sign of the real part of the Fourier transform of template g.

$$G_{BPOF}(\omega_\rho, \omega_\theta) = e^{-j\Phi_{BPOF}(\omega_\rho, \omega_\theta)} \tag{4.9}$$

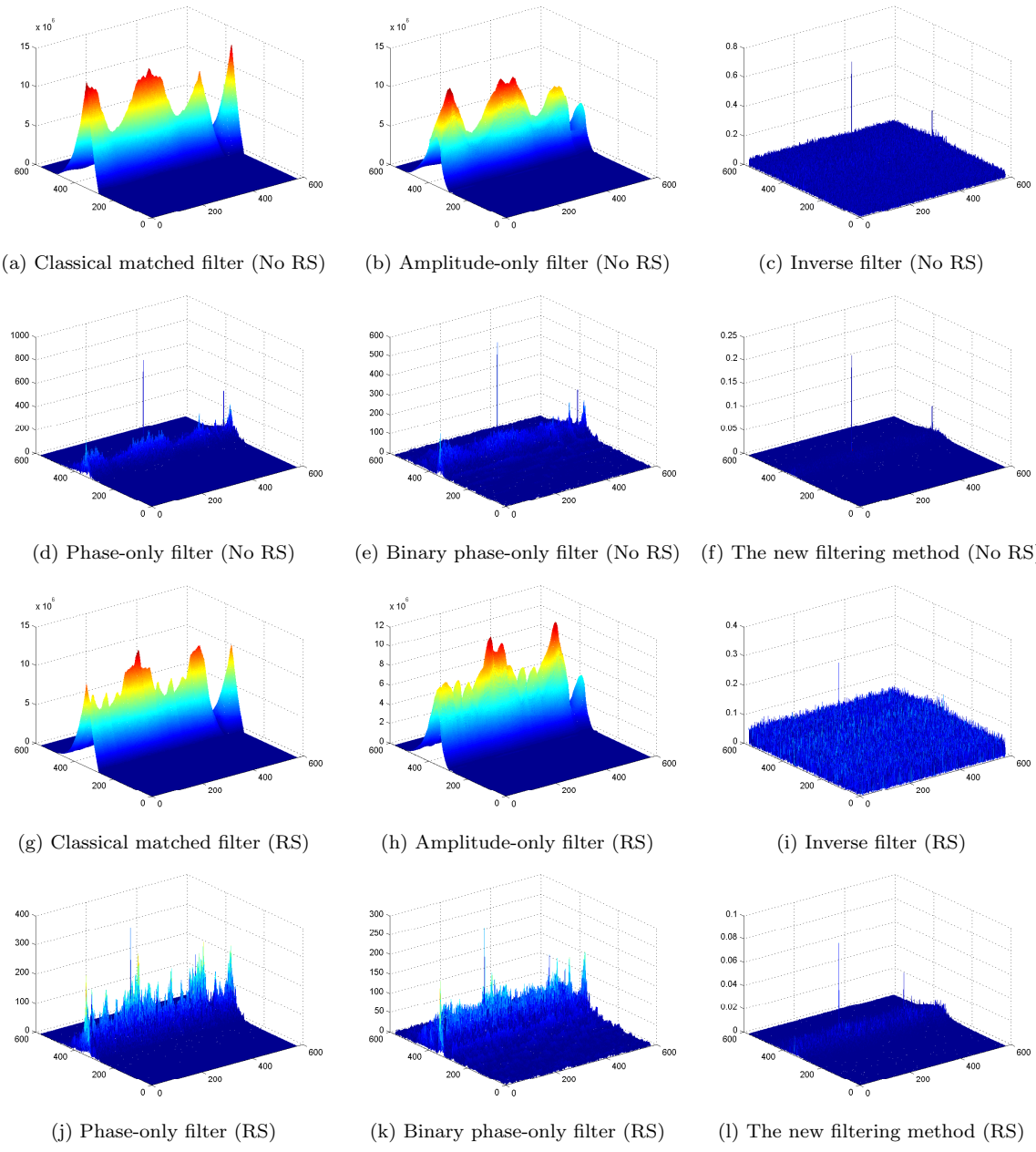
where

$$\Phi_{BPOF}(\omega_\rho, \omega_\theta) = \begin{cases} 0^\circ & G_r \geq 0 \\ 180^\circ & G_r < 0 \end{cases} \tag{4.10}$$

and G_r stands for the real part of the Fourier transform $G(\omega_\rho, \omega_\theta)$.

Phase information is considerably more important than the amplitude information in preserving the visual intelligibility of an image. A phase-only filter (POF) is obtained by setting the magnitude to unity for all the frequencies. The advantages of the POF are: 1) all the energy gets through the filter plane without magnitude function's attenuation; 2) the energy is concentrated in a much narrower peak with better discrimination capability; 3) the POF retained only the phase information requires less space to store the data [101]. Normally, the phase-only filter yields much sharper correlation peaks and better discrimination [34]. In the recent literature, phase information was applied in optical signal processing [28] [101], image registration [102], broadcasting [103], and digital image watermarking [29].

However, according to the experimental results illustrated in Fig. 4.4, all these five types of traditional filters, including the phase-only filter, fail to produce acceptable discrimination when rotation or scaling or both applied to the watermarked image. These figures show the cross-correlation results in 3-dimensions between I-frame of one GOP from the watermarked video *Mobile* in LPM domain and a template cut from this domain. Different filters are applied to calculate the matching results. "No RS" means no RST attacks applied to the target frames; "RS" means the target frames are rotated by 20° and scaled to 70%. The classical matched filter and the amplitude-only filter have unacceptable discrimination (refer to Fig. 4.4 (a), (b)). The inverse filter,



(a) Classical matched filter (No RS) (b) Amplitude-only filter (No RS) (c) Inverse filter (No RS)

(d) Phase-only filter (No RS) (e) Binary phase-only filter (No RS) (f) The new filtering method (No RS)

(g) Classical matched filter (RS) (h) Amplitude-only filter (RS) (i) Inverse filter (RS)

(j) Phase-only filter (RS) (k) Binary phase-only filter (RS) (l) The new filtering method (RS)

Figure 4.4: Matching results by using cross-correlation with different filters.

the phase-only filter and the binary phase-only filter perform reasonably well when there is no rotation or scaling applied to the watermarked frame. The sharp peaks in Fig. 4.4 (c), (d) and (e) clearly indicate the position of the template in the LPM of the watermarked image. However, if the watermarked image has undergone rotation or scaling transform, the four filters (classical filter, amplitude-only filter, phase-only filter and binary phase-only filter) fail to produce acceptable discrimination (refer to Fig. 4.4 (g), (h), (j), and (k)). Only the inverse filter among these five traditional filters has a possibly sufficient discrimination (refer to Fig. 4.4 (i)). There are higher values in the side area beside the peak. If we add more attacks, such as compression, the peak will disappear.

Since all types of traditional filters fail to produce acceptable discrimination while rotation or scaling or both applied to the watermarked image, we devised a new filtering method to deal with this problem.

4.3.2 Phase correlation

Before we conduct our filtering method, we need to introduce phase correlation as the background theory.

The phase correlation is an efficient approach to rectify the watermark position to avoid exhaustive search [104]. Kuglin and Hines proposed the phase correlation based on the shift theorem of the Fourier transform [105]. Given an image f_0 and its shifted version f_1 with the displacement (x_0, y_0), i.e.,

$$f_1(x, y) = f_0(x - x_0, y - y_0) \tag{4.11}$$

Then, the relationship between their corresponding Fourier transforms F_0 and F_1 is

as follows:

$$F_1(u,v) = e^{-j(ux_0+vy_0)}F_0(u,v) \tag{4.12}$$

We compute the cross-power spectrum of the two images defined as

$$C = \frac{F_1(u,v)F_0^*(u,v)}{|F_1(u,v)F_0^*(u,v)|} = e^{-j(ux_0+vy_0)} \tag{4.13}$$

where F^* is the complex conjugate of F. The shift translation property guarantees that the phase of the cross-power spectrum is equivalent to the phase difference between the images. Furthermore, we represent the phase of the cross-power spectrum in its spatial form, i.e., by taking the inverse Fourier transform of the representation in the frequency domain,

$$D = \mathfrak{F}^{-1}(C) \tag{4.14}$$

where $\mathfrak{F}^{-1}$ is the inverse Fourier transform.

Based on the property of the Fourier transform, the Fourier transform of impulse function $\delta(x-d)$ is e^{-jwd}. Equ. (4.14) gives a two-dimensional δ function centered at the displacement. So D is a function which is an impulse, that is, it is approximately zero everywhere except at the displacement.

4.3.3 Template cross-correlation

We use a template, which is cut from the image, to calculate the cross-correlation between the template and the image, given an image $f(x, y)$ with the size of $N \times N$ and a template $t(x, y)$ with the size of $(x_0 \times y_0)$.

First, we define a rectangle window.

$$w = \begin{cases} 1 & 0 < x < x_0, 0 < y < y_0 \\ 0 & else \end{cases} \tag{4.15}$$

Then, the template $t(x, y)$ cut from this image can be described as follows,

$$t(x, y) = w. * f(x - x_0, y - y_0) \tag{4.16}$$

$t(x, y)$ can be simplified as follows:

$$t(x, y) = f(x - x_0, y - y_0) - f'(x - x_0, y - y_0) \tag{4.17}$$

where,

$$f'(x - x_0, y - y_0) = (Aw - w). * f(x - x_0, y - y_0) \tag{4.18}$$

Aw is an all-one window to cover the whole image.

According to Equ. (4.3), the correlation between $t(x, y)$ and $f(x, y)$ can be calculated as follows:

$$cor = \mathfrak{F}^{-1}[T(u, v) \cdot F^*(u, v)] \tag{4.19}$$

where, $T(u,v)$ and $F(u,v)$ are the DFT of $t(x,y)$ and $f(x,y)$ respectively. $T(u,v)$ can be described as:

$$T(u,v) = F(u-x_0, v-y_0) - F'(u-x_0, v-y_0) \tag{4.20}$$

Then, the correlation can be written as follows:

$$\begin{aligned} cor &= \mathfrak{F}^{-1}[T(u,v)\cdot F^*(u,v)] \\ &= \mathfrak{F}^{-1}[F(u-x_0,v-y_0)\cdot F^*(u,v) - F'(u-x_0,v-y_0)\cdot F^*(u,v)] \end{aligned} \tag{4.21}$$

For the cross-correlation, $F'(u-x_0, v-y_0)$ can be considered as noise. Therefore, the second part $F'(u-x_0,v-y_0)\cdot F^*(u,v)$ can be approximately considered as zero. The first part $F(u-x_0,v-y_0)\cdot F^*(u,v)$ contributes most to the results. If we use only phase information for both F and F^*, the correlation can be written as follows:

$$\begin{aligned} cor &= \mathfrak{F}^{-1}[T(u,v)\cdot F^*(u,v)] \\ &= \mathfrak{F}^{-1}[F(u-x_0,v-y_0)\cdot F^*(u,v) - F'(u-x_0,v-y_0)\cdot F^*(u,v)] \\ &\approx \mathfrak{F}^{-1}[F(u-x_0,v-y_0)\cdot F^*(u,v)] \\ &\approx \mathfrak{F}^{-1}[\frac{F(u-x_0,v-y_0)\cdot F^*(u,v)}{|F(u-x_0,v-y_0)\cdot F^*(u,v)|}] \\ &\approx \mathfrak{F}^{-1}[e^{j(ux_0+vy_0)}] \end{aligned}$$

Therefore, we get the same results as D in Equ. (4.14), which is a two-dimensional impulse function centered at the displacement.

4.3.4 Phase-only filtering method

From detection theory, correlation detectors are optimum for a Linear Time Invariant (LTI), frequency non-dispersive, Additive White Gaussian Noise (AWGN) channel [106]. However, usually, we use real images as our experimental targets, whose power spectrum is not white. It is still possible to achieve optimum detection for non-white Gaussian noise, by applying a so-called whitening filter at the input of the correlation receiver. This filter transforms the non-white input signal of the receiver to a signal with a constant power spectrum [107].

From the theory we get from above sections, we propose a new filtering method, named phase-only filtering method, to transform the non-white spectrum in the LPM domain of a frame to a mapping with an unity power spectrum, and then apply phase-only filter to the resulting mapping.

The filtering process is described as follows:

$$r = \mathfrak{F}^{-1}[F_{\Phi}(w_{\rho}, w_{\theta}) \cdot G^{*}_{\Phi}(w_{\rho}, w_{\theta})] \tag{4.22}$$

where

$$F_{\Phi}(w_{\rho}, w_{\theta}) = e^{-j\Phi_{F}(w_{\rho}, w_{\theta})} \tag{4.23}$$

and $\mathfrak{F}^{-1}$ is the inverse DFT. The new filtering method gives extremely sharp peaks no matter whether rotation and/or scaling are applied to the watermarked image or not (refer to Fig. 4.4 (f) and (l)). It clearly shows the position of the template when other filters could not.

4.4 Watermark embedding

Video clips have huge information comparing with images. We conduct the watermark embedding and detection in the I-frame of each GOP. We embed the watermark sequence into the log-polar mapping of each I-frame. In our implementation, the watermark sequence is same for each I-frame. However, it can be different from one to the other. The cost of the key will be increased as consequence. The advantage of this scheme is that we find the watermark position directly in the LPM domain of the frame instead of the geometrical rectification of the frame in the spatial domain to avoid the imprecisions introduced by the interpolation during the log-polar mapping.

The procedure of embedding a watermark consists of the following steps (refer to Fig. 4.5):

1. Use a PN generator to generate a watermark data sequence, which is spread spectrum consisting of both positive and negative values.

2. Divide the target video clips to Group of Pictures (GOP). Here, we fix the number of pictures of GOP to 8.

3. Truncate I-frame to a smaller square image. The size of our target video is 704×576 or 704×480. The truncated part size is 576×576 or 480×480 respectively.

   The reason why we truncate an image to a square image is that the square image is more tolerant to rotation attack in LPM domain than the rectangle image. A square image always has less loss with rotation and cropping than a rectangular image. We calculate the rotation loss as the percentage of zero-padding after rotation and cropping of the target image. Figure 4.6 shows the rotation loss for a square image with a size of 576×576, a bigger rectangular image with a

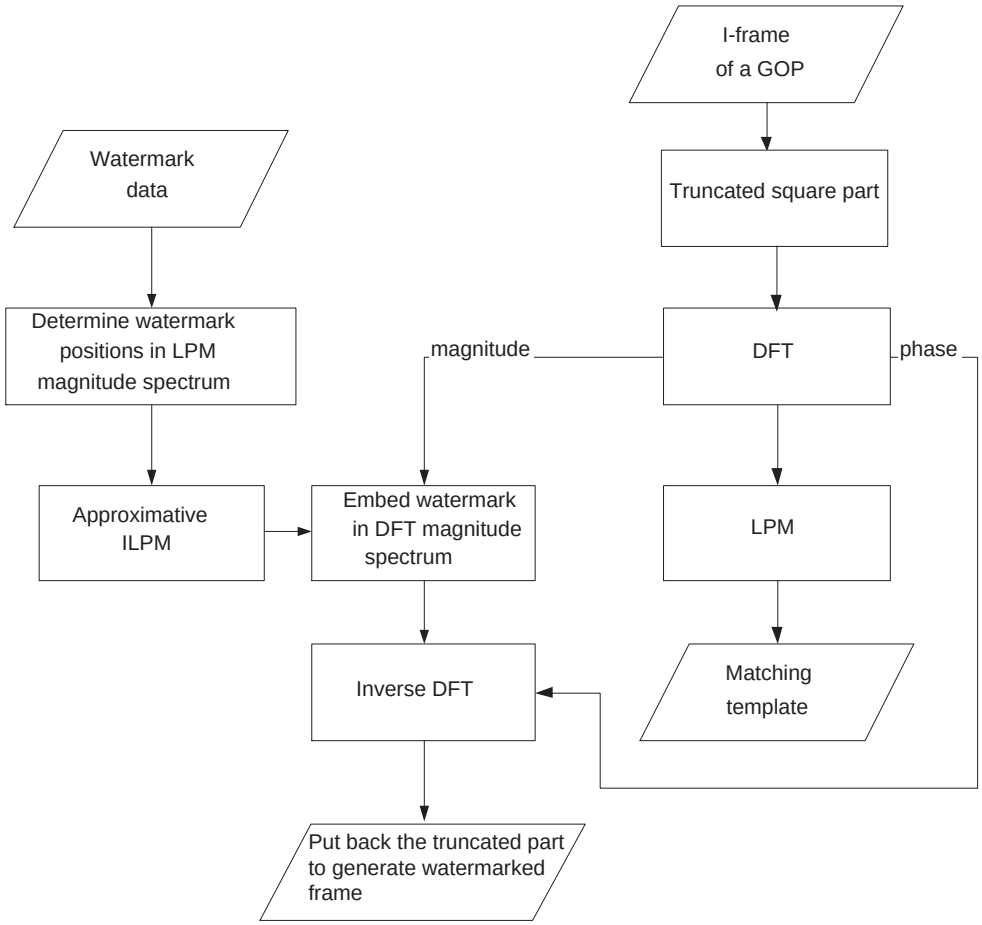


Figure 4.5: Watermark embedding.

size of 704×576 and a smaller rectangular image with a size of 576×384. Both rectangular images have higher loss than the square image after rotation and cropping. The biggest gap between them is around 45°, where the square image has 16.91% loss, the bigger rectangular image has 18.15% loss, and the smaller rectangular image has 21.78%.

4. Compute DFT of the truncated image from I-frame of each GOP. The magnitude spectrum of the DFT is positive, so we need to extend the watermark data sequence to its double length, i.e. encode positive numbers x as $(x, 0)$ and negative

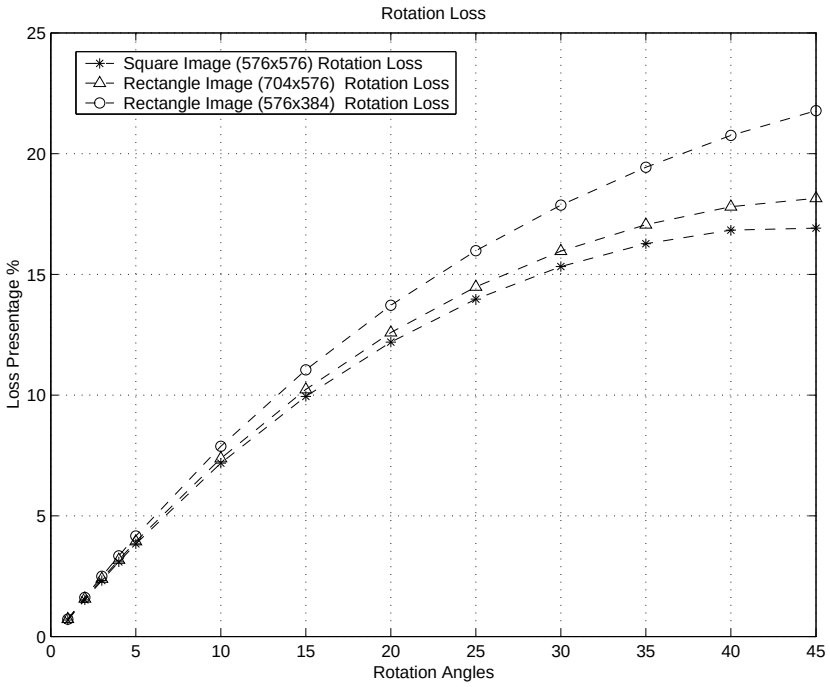


Figure 4.6: Rotation loss.

numbers x as $(0, x)$.

5. Save a block portion of LPM spectrum as the matching template g, which will be available to the watermark extraction process.

6. Select the desired locations in the LPM magnitude spectrum for embedding the watermark data sequence.

7. Embed the watermark in the approximate ILPM domain. If we embed the watermark in the LPM domain, we need ILPM to transform back from the LPM domain to the DFT domain. To avoid computing ILPM that may bring unacceptable computational imprecision, we use the approximate ILPM and embed the watermark in DFT domain. The watermarking locations in the Cartesian DFT magnitude spectrum is approximated from the watermark points in the LPM

domain selected in step 6.

Naturally, if we want to change the value of one point in LPM magnitude spectrum for embedding watermark data, we only need to find the corresponding four points in Cartesian magnitude spectrum and change their values accordingly. For details, refer to [104].

Most algorithms use a simple addition to embed watermark, refer to Equ. (4.24),

$$E' = E + \beta * W \tag{4.24}$$

where E is the DFT amplitude spectrum of the original image, W is the watermark data, E' is the modified DFT amplitude spectrum of the original image, and β is watermarking strength used to achieve the tradeoff between the robustness and the visibility of the watermark.

If the difference between E and $\beta * W$ is small enough, we can use Equ. (4.25) to replace the values at the embedding points by $\beta * W$.

$$E' = \beta * W \tag{4.25}$$

This embedding process will not change the magnitude values of those embedding points dramatically, therefore the goal of invisibility can be achieved. Meanwhile, the embedding method can simplify the extraction process.

8. Apply inverse DFT to get the watermarked image. Put back the watermarked image into its original frame as the correct location during truncating.

9. Generate the watermarked video together with other frames.

During the embedding process, the symmetry of the DFT magnitude spectrum should be maintained, thus we must carefully select the desired points in the LPM magnitude spectrum.

To get the tradeoff between invisibility and robustness, we choose the middle frequency components as the location to insert watermark data. Because the log-polar mapping is just like a sampling process, the closer to the center, the higher the sampling rates. If we insert the watermark data into the low frequency components, the change of the value of one point in the Cartesian magnitude spectrum will cause value changes of a lot of points in the log-polar magnitude spectrum because of the bilinear interpolation. That may cause imprecision in the extraction process. Therefore, in our watermarking algorithm, we insert the watermark data into the middle frequency components. The experimental results show the effectiveness of this approach.

4.5 Watermark detection

To the watermark extraction process, available are the watermarked video that may or may not suffered from attacks, the watermarking key, the original watermark positions, and the matching template g. For each GOP, the watermarking key could be same. However, the original watermark positions and the matching template would be different since the watermark positions will be the adaptively selected according to the individual situation of different frames to get the optimal fidelity and robustness. The procedure of extracting the watermark consists of following steps (refer to Fig. 4.7):

1. Apply DFT and LPM to each truncated I-frame of the watermarked video, and transform it into LPM domain.

2. Calculate the correlation between the LPM domain of the watermarked frame

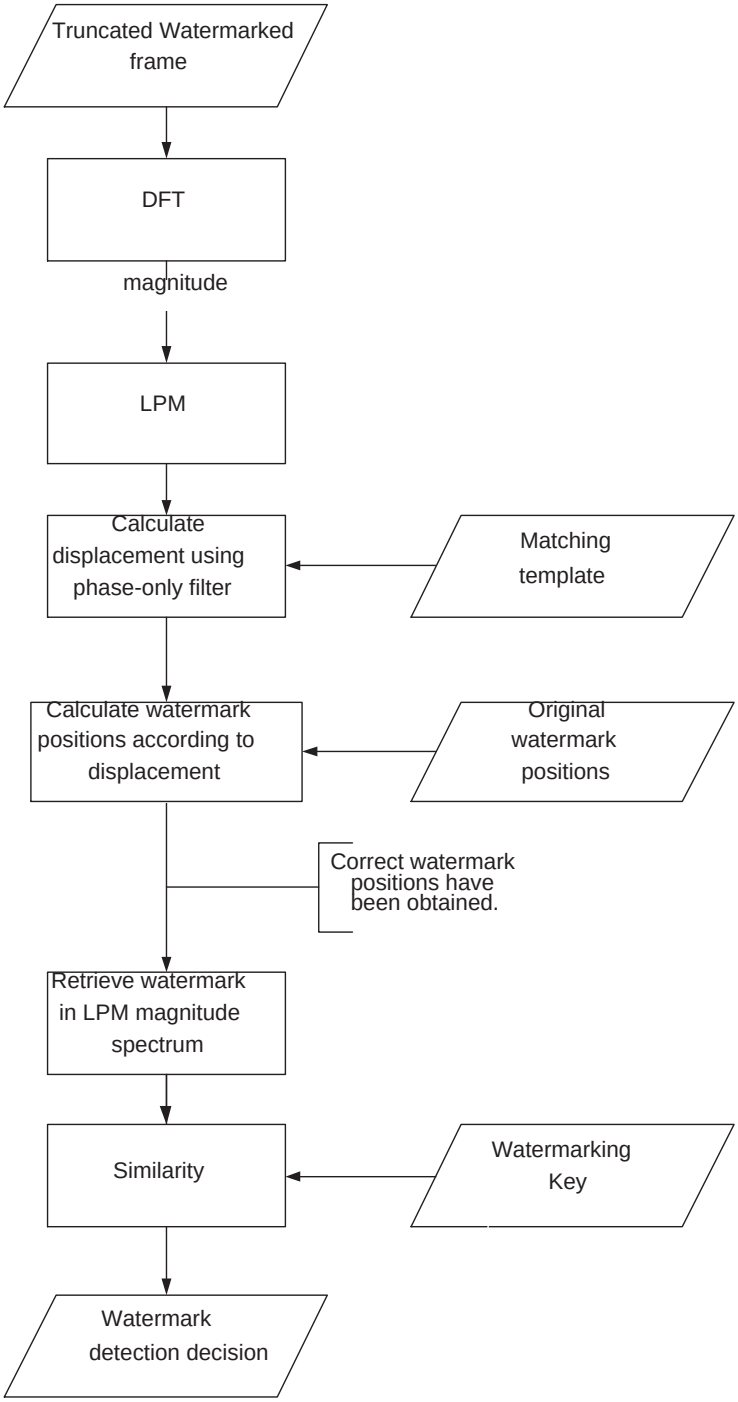


Figure 4.7: Watermark detection.

and the matching template g, by using the new filtering method, according to Equ. (4.22). For details, refer to Section 4.3.

3. Find the peak in the calculated correlation, and find the translations according to Equ. (5.1). Suppose the coordinates of the original watermark position for w_i is $(\rho_{w_i}, \theta_{w_i})$, then we can calculate the coordinates of the new watermark position in the watermarked image having undergone attacks by using the following equations:

$$\rho_{w_i}{}' = \rho_{w_i} + \Delta_\rho \tag{4.26}$$

$$\theta_{w_i}{}' = \theta_{w_i} + \Delta_\theta \tag{4.27}$$

4. Retrieve the watermark data V at the rectified location by using the following equation:

$$V = \frac{E''}{\beta} \tag{4.28}$$

where E'' is the DFT magnitude spectrum of the watermarked frame that undergone RST transformations and other attacks, while β is defined in Equ. (4.24). The value change of V caused by a scaling transformation is proportional to the scale factor, and the normalized correlation calculated by Equ. (4.29) is independent of the change.

The scaling operation will change the value of the DFT magnitude spectrum, which is proportional to the scaling factor. The correlation function can be normalized for magnitude changes by using the normalized correlation, which is in the range of -1 to 1, independent of scale changes in the magnitude (refer to Equ. (4.29).

5. By using Equ. (4.29), calculate the normalized correlation coefficient between the original watermark data and the extracted watermark data. 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.

$$sim = \frac{W \times V^T}{\sqrt{(W \times W^T)(V \times V^T)}} \tag{4.29}$$

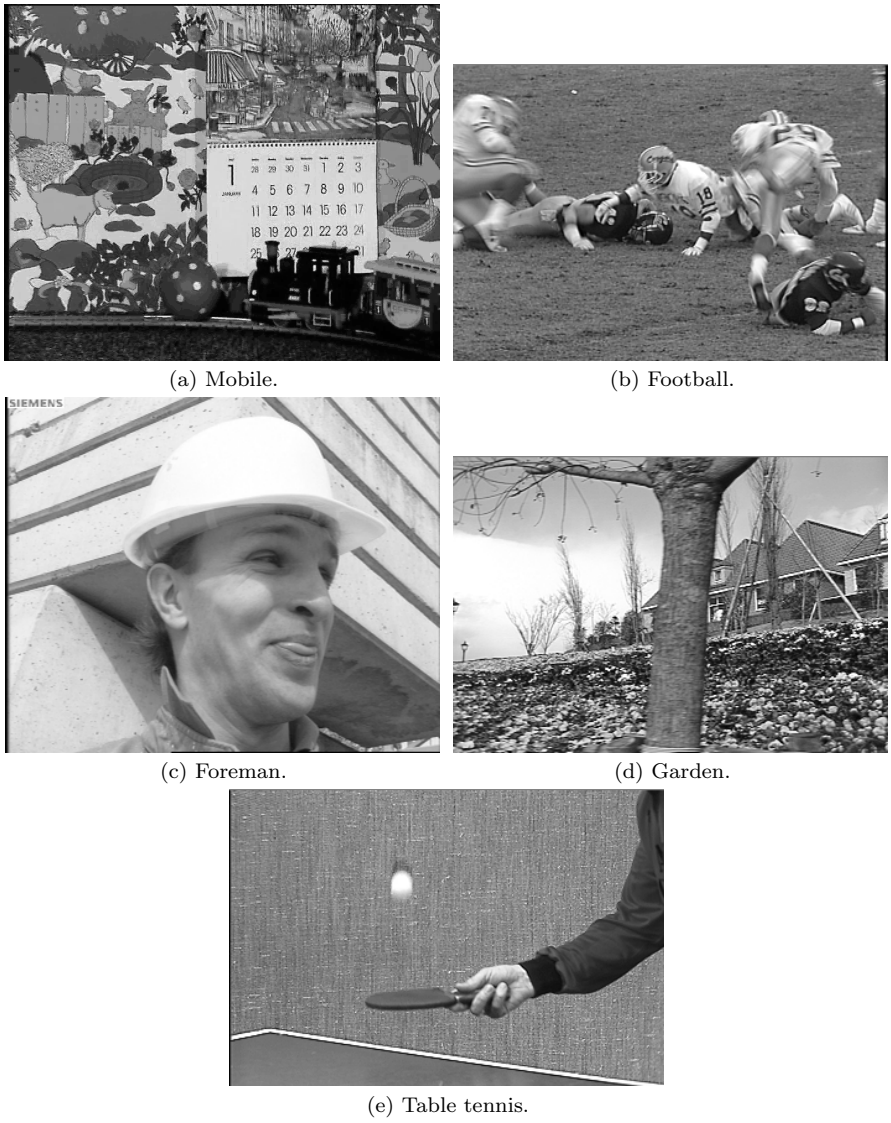
where W and V are, respectively, the original watermark vector and retrieved watermark vector, and $(\cdot)^T$ is the transpose operation of a matrix.

4.6 Experimental results

In this section, we will illustrate the performance of the proposed algorithm. The target videos we tested for our algorithm are shown in Fig. 4.8. We use 5 test videos, and their sizes are shown in Table 4.1. In the Table 4.1, "T", "S" and "R" present "Translation", "Scaling" and "Rotation", respectively.

4.6.1 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 4.1. Human eyes could not recognize the difference between the original video and the watermarked video under these PSNR's.



(a) Mobile.

(b) Football.

(c) Foreman.

(d) Garden.

(e) Table tennis.

Figure 4.8: Test videos.

Table 4.1: Similarity results for target videos

	Mobile 704 × 576 (4CIF)		Football 704 × 480 (4SIF)		Table tennis 704 × 480 (4SIF)		Foreman 704 × 576 (4CIF)		Garden 704 × 480 (4SIF)	
	Mark	No mark	Mark	No mark	Mark	No mark	Mark	No mark	Mark	No mark
PSNR	39.8488	-	39.4482	-	39.1569	-	39.8103	-	40.3453	-
T	0.9469	0.1269	0.9943	0.1231	0.9438	0.2121	0.9833	0.1256	0.9494	0.1522
S 70%	0.8609	0.2215	0.7876	0.2835	0.8813	0.1390	0.8704	0.1543	0.7199	0.3007
80%	0.8893	0.2351	0.8602	0.2618	0.8975	0.0537	0.8527	0.1565	0.7509	0.2504
90%	0.8692	0.0547	0.9692	0.1890	0.9139	0.0565	0.8676	0.1211	0.8552	0.2012
110%	0.8533	0.1353	0.9745	0.2779	0.9033	0.0456	0.9399	0.1998	0.9311	0.1601
120%	0.7957	0.1246	0.9621	0.1934	0.9513	0.1471	0.9321	0.1835	0.9164	0.1817
130%	0.7432	0.1492	0.9653	0.3150	0.8919	0.1459	0.7987	0.1754	0.7284	0.3521
R 0°	0.9546	0.0934	0.9975	0.2454	0.9705	0.2068	0.9956	0.0234	0.9805	0.3118
1°	0.7693	0.1402	0.8984	0.2657	0.8656	0.1832	0.8996	0.1243	0.8447	0.3592
2°	0.8419	0.3247	0.9182	0.2860	0.8234	0.2052	0.8734	0.1298	0.8476	0.3092
3°	0.8438	0.2553	0.8883	0.1775	0.8982	0.2082	0.8654	0.2122	0.7968	0.2921
4°	0.8255	0.2949	0.9094	0.2198	0.9045	0.3501	0.8453	0.1987	0.8094	0.2727
5°	0.8800	0.1237	0.8751	0.2287	0.8489	0.1638	0.8939	0.1148	0.8058	0.3086
10°	0.7994	0.2674	0.9046	0.2986	0.8642	0.1463	0.8658	0.0374	0.8226	0.4210
15°	0.8667	0.2389	0.8635	0.2617	0.8125	0.1897	0.8625	0.0917	0.8620	0.2582
20°	0.7885	0.2046	0.9206	0.3660	0.8179	0.1505	0.8459	0.2894	0.8071	0.1836
25°	0.8344	0.2612	0.9245	0.3018	0.8428	0.1109	0.7872	0.2655	0.8521	0.2760
30°	0.8123	0.2762	0.8542	0.4346	0.7299	0.2700	0.8647	0.2945	0.7605	0.3412
35°	0.7803	0.2259	0.9173	0.1105	0.8726	0.2016	0.8467	0.2835	0.7234	0.3067
40°	0.6878	0.2071	0.9021	0.1694	0.8078	0.3387	0.7975	0.3087	0.7035	0.3234
45°	0.5450	0.0976	0.8991	0.2334	0.6719	0.1682	0.8157	0.2150	0.6054	0.2588
RST	0.7565	0.2855	0.8884	0.3179	0.7360	0.2779	0.7608	0.2321	0.7209	0.2522
Noise	0.8742	0.0958	0.8390	0.3321	0.8648	0.1564	0.8521	0.1908	0.8456	0.2398
LP	0.9574	0.0774	0.9963	0.1969	0.9608	0.2090	0.9210	0.1987	0.9209	0.1654

4.6.2 Rotation with cropping

Normally, the rotation is very slight for video signals. The rotation angles are not more than 5°. Table 4.1 shows the experimental results for rotated watermarked and unwatermarked videos up 45°. In this table, "Mark" means the results for watermarked videos and "No mark" means the results for unwatermarked videos. The similarity values for watermarked video is much higher than the ones for unwatermarked video. Even up to 45°, the similarity values for watermarked video and the values for unwatermarked video can still be successfully separated. Table 4.1 suggests that the proposed algorithm is robust to rotation.

Table 4.2: MPEG2 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	17.89	18.09	22.85	23.45	20.85	21.98	18.49	19.02	23.35	24.89
MPEG2	0.5909	0.2651	0.5403	0.2232	0.5422	0.2032	0.5576	0.1866	0.5210	0.2087
PSNR	21.32	22.55	25.46	26.66	23.74	24.78	22.53	23.43	26.49	27.09
+ RST	0.5012	0.2321	0.5332	0.1879	0.5033	0.2037	0.5287	0.2036	0.5020	0.2104

4.6.3 Scaling

Here, we consider scaling ratios from 70% to 130%. We use "bilinear" interpolation method to resize each frame to the scaled size. In Table 4.1, The similarity values for watermarked video is much higher than the ones for unwatermarked video. We could successfully tell whether a video is watermarked or not. Therefore, we could suggest that the proposed algorithm is robust to scaling.

4.6.4 Translation

The results for translation are shown in Table 4.1. No matter how many pixels the translations happened to the watermarked video are, the similarity results for watermarked video are the same after re-synchronization.

4.6.5 RST combination attacks

In Table 4.1, RST means the combination attack of rotation by 15°, scaling by 110%, and translation by 50 pixels. The scheme is robust against this attack.

4.6.6 Noises and filtering

We checked the results for additive Gaussian noise and low-pass filtering. We can show from the data in Table 4.1 that our scheme is robust to these attacks. In Table 4.1,

"Noise" means Gaussian noise, and "LP" means low-pass filtering.

4.6.7 MPEG-2 compression

We applied MPEG-2 to our watermarked video and tested the worst cases that our algorithm could robust to. In Table 4.2, PSNR shows the quality of the watermarked video after MPEG-2 compression. With the bad quality of the compressed video and PSNRs are around 20 dB, our scheme still could successfully detect the watermark.

4.6.8 MPEG-2 with RST attacks

Both MPEG-2 and RST attacks are applied to the watermarked video here. Table 4.2 shows the simulation results in the row "+ RST". In order for the algorithm to work at these attacks, the quality of the compressed video has to be with higher PSNR than in Section 4.6.7. The simulation results prove that our algorithm is robust against these attacks.

4.6.9 MPEG-2 recoding

In this algorithm, we embed the watermark into I-frame of each GOP. In our implementation, we fix the number of frames in one GOP in MPEG-2 compression parameter list. This will bring us a potential issue that attackers could re-encoding and decoding with different GOP numbers. In that way, the pre-watermarked I-frame would become P-frame or B-frame, which have more compression ratio. We implement this situation with different GOP number in MPEG-2 re-encoding and decoding. According to our implementation results, the similarity value of P-frame watermark embedding is only about 40% of the value of I-frame embedding. The similarity result of B-frame wa-

termark embedding is even worse, which is only around 26% of the value of I-frame embedding.

In order to solve this problem, we could embed watermark into each frame of GOP. The benefit of it is the robustness to MPEG-2 recoding because I-frame always contains watermark no matter how the GOP is defined. The disadvantage is the computational cost.

4.7 Discussions

This video watermarking algorithm based on the log-polar mapping and phase-only filtering method is extended from our image watermarking algorithm [100]. Working with I-frame of each GOP, the advantage of this method is very robust against rotation, scaling and translation attacks, even with additive noise, low-pass filtering and MPEG-2 compression. For example, video rotation is normally very slight, less than 5°. With small rotation angles, the similarity values could be higher than 0.8, which shows the excellent robustness to rotation.

However, the detection method fails for the most advanced video compression standard - H.264. When the watermarked video is compressed with H.264 with a similar PSNR as compressed by MPEG-2, the similarity results for the watermarked video having undergone H.264 compression are only around or less than 0.3. The watermark pattern existence cannot be determined after H.264 compression because H.264 compression destroys watermark structure with more efficient compression of I-frame. This scheme relies on I-frames of each GOP to carry the watermark load.

A template is needed for watermark location synchronization for each I-frame and it has to be transmitted to the receiver side along with the watermark key. It is an

extra cost for the watermark scheme besides the watermark pattern.

These motivated us to develop a novel video watermarking algorithm to be robust against geometric attacks and H.264 compression. We will introduce the novel video watermarking algorithm in Chapter 6.

Chapter 5

Rectification of RST transformations for video watermarking algorithms

We have introduced a video watermarking algorithm (LPMPOF), which is robust against RST distortions and MPEG-2 compression based on log-polar mapping and phase-only filtering method in Chapter 4. The watermark embedding and detection of the algorithm are conducted in the LPM domain. The normalized cross-correlation is used to detect the existence of watermark in the target signals.

Furthermore, the algorithm is capable of detecting the parameters of rotation, scaling and translation that the watermarked signal have undergone. Therefore, the algorithm can work together with any other watermarking algorithms as a RST parameters detector. Fig. 5.1 shows the methodology of the integration of LPMPOF algorithm with other algorithms, in which LPMPOF algorithm, as a RST parameter detector, works independently from the watermark embedding and detection of a foreign water-

marking algorithm. Watermarked video and the templates are the input to the RST parameter detector, and the output is RST parameters, which are used to rectify the watermarked video having undergone RST distortions before the watermark detection.

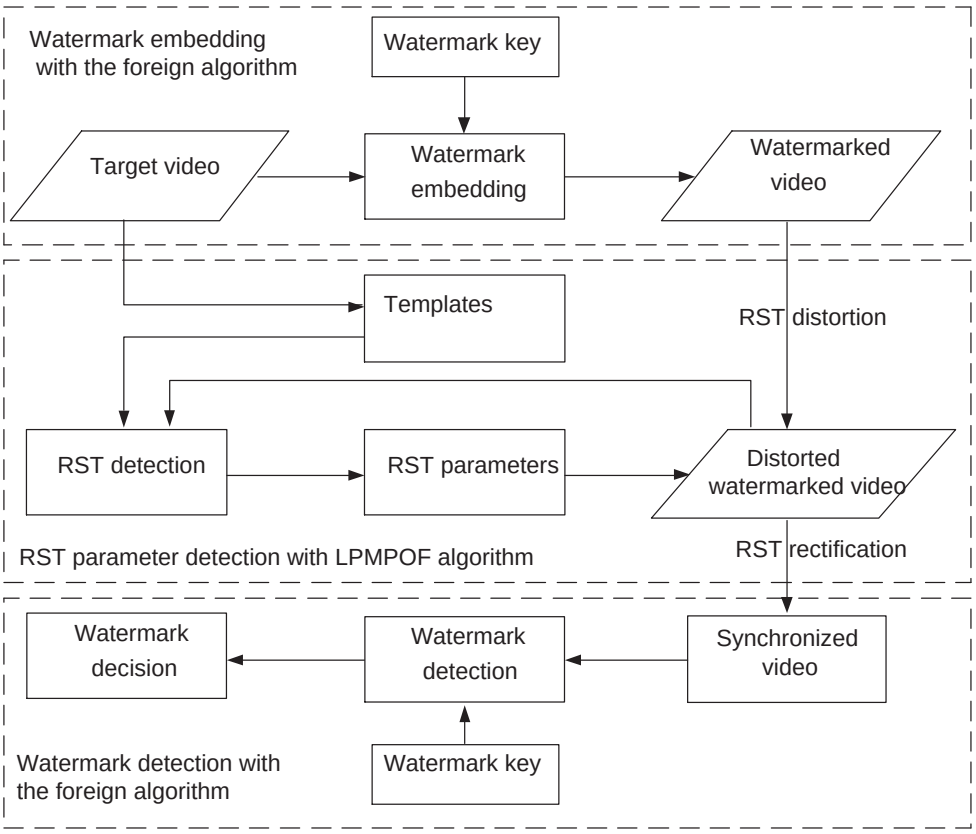


Figure 5.1: Integration of LPMPOF algorithm with other algorithms.

In this chapter, to demonstrate the usefulness of our RST invariant watermarking algorithm, we will integrate our algorithm into a semi-fragile algorithm, DCTLSB algorithm [108], to enhance the robustness to RST distortions.

5.1 Motivation

The LPMPOF algorithm carries watermark message and to be robust against RST attacks. However, the capacity is probably only one bit because the watermark detector only declares whether there is a watermark by checking the similarity results against a predefined threshold in each GOP of the watermarked video. Therefore, we can utilize the best feature of LPMPOF algorithm to detect the RST parameters and use other watermarking algorithms to carry more watermark load as watermark message carrier.

The DCTLSB algorithm is a semi-fragile watermarking algorithm, in which a watermark is effected only by illegitimate distortions. In other words, the watermarking algorithm can be robust against various attacks to a certain extent. Beyond this point, the watermark is distorted and cannot be detected. The DCTLSB algorithm can be robust against quantization, low-pass filtering, and noise addition to some extent. However, it will fail to detect watermark with even very slight RST distortions due to the loss of synchronization. The algorithm itself cannot re-synchronize. However, it has the high capacity. This motivates us to integrate our LPMPOF algorithm with the DCTLSB algorithm to generate both high capacity and robustness to RST distortions.

5.2 RST parameters detection

As we mentioned in Chapter 4, the LPMPOF watermarking algorithm is working in the LPM domain, which is the log-polar mapping of the magnitude of Fourier transform of I-frame of a GOP in the target video. Rotation and scaling in the spatial domain are turned to be translational shift in the LPM domain. Translation has no affect in the LPM domain. This algorithm has strong robustness to RST distortions because it can detect and compute the exact rotation angle, scaling ratio and translation parameters

for the target video. The technique is based on our phase-only filtering method, which could give the best discrimination for the cross-correlation between the template and the target frame comparing with other traditional filters. This important feature makes it possible to use the LPMPOF algorithm as a RST parameters detector for other video watermarking algorithms.

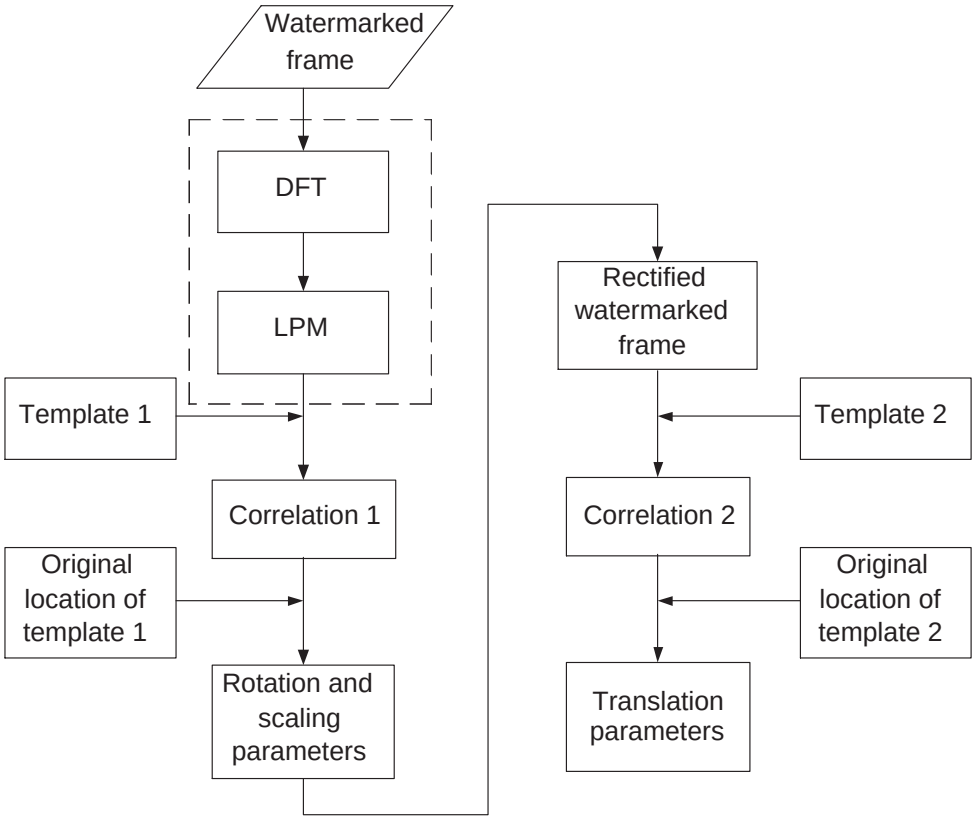


Figure 5.2: RST parameters detection.

5.2.1 Calculation of rotation and scaling parameters

As shown in Fig. 5.2, $Template1$ is used in the LPM domain to detect the current watermark location and furthermore to obtain the rotation and scaling parameters. The peak in the correlation spectrum is the position of the matching template in the

LPM spectrum of the watermarked frame having undergone RST attacks. Suppose the coordinate of the peak is (ρ_1, θ_1), and we know the original position of $Template1$ is (ρ_0, θ_0), so the translations in the LPM domain are:

$$\begin{cases} \Delta_\rho &= \rho_1 - \rho_0 \\ \Delta_\theta &= \theta_1 - \theta_0 \end{cases} \tag{5.1}$$

where Δ_ρ and Δ_θ are the number of pixels shifted along the ρ axis and the θ axis respectively. They correspond with the rotation and scaling parameters, respectively, in the spatial domain. Therefore, the rotation and scaling parameters can be calculated by using Equ. (5.2).

$$\begin{cases} \alpha' &= \frac{360^0 \cdot \Delta_\theta}{N'} \\ \sigma' &= e^{\frac{\ln(r_{max}) \cdot \Delta_\rho}{M'}} \end{cases} \tag{5.2}$$

with

$$r_{max} = \sqrt{(M'/2)^2 + (N'/2)^2} \tag{5.3}$$

where, M' and N' are, respectively, the number of pixels along the ρ-axis and θ-axis in the LPM domain ($M' \times N'$ is the size of the LPM domain), and $M \times N$ is the size of the magnitude spectrum of the Fourier transform. $M' \times N'$ may be different with $M \times N$ because the sampling points can be increased or decreased during the log-polar mapping. α' and σ' are, respectively, the rotation angle and scaling ratio in the spatial domain calculated from Δ_θ and Δ_ρ in the LPM domain. Therefore, we can invert the scaling and rotation transforms to synchronize the watermarked frame geometrically.

5.2.2 Detection precision of rotation and scaling parameters

The detection precision of rotation and scaling parameters depends on two factors. The first one is the filter matching accuracy between the template and distorted domain where the template cut from. The second factor is the number of sampling points along both θ and ρ axis. As illustrated in Chapter 4, the phase-only filtering method is able to indicate clearly the location of the template in the LPM domain, which guarantees the first factor is satisfied.

For the second factor, we have to consider the trade-off between the computation precision and complexity. The rotation parameter calculation is shown in Equ. (5.1). If the one pixel is shift, which is presented by $\Delta_\theta = 1$, the bigger the number of sampling points N', the smaller rotation angle α'. For example, with $N' = 512$, the maximum error for rotation angle α' could reach $360°/512 = 0.7031°$. For $N' = 1024$, the error for α' is always less than $360°/1024 = 0.3516°$. For $N' = 2048$, the maximum error for α' can be $360°/2048 = 0.1758°$. However, if the sampling points are higher, the computation complexity of log-polar mapping will be higher and the implementation will become time-consuming. Therefore, we choose 1024 as the number of sampling points for θ in our implementation and the precision results are shown in Table 5.1.

The relationship between the detection precision of the scaling factor σ and the sampling points are not straightforward as rotation angles. We test the detected scaling factor with three different sampling points combination on θ and ρ, shown in Table 5.2. "Detected scaling factor with $M \times N$" in the table shows the detected scaling factor with the different combination of numbers of sampling points on horizontal and vertical direction. M represents the number of sampling points for ρ axis and N is the number of sampling points for θ axis. For example, "Detected scaling factor with 1024×1024" in Table 5.2 shows the detected scaling factors calculated with 1024 sampling points

on both ρ and θ axis. Three detected scaling factors under different numbers of sampling points are quite similar, shown in Table 5.2. Considering the computational complexity, we decide to use the sampling points of 512 for ρ and 1024 for θ in our following implementation.

Table 5.1: Rotation parameter detection precision with the sampling points of 1024 or 512 on θ

Applied angle	Detected angle with 1024 sampling points	Detected angle with 512 sampling points
1°	1.0547°	1.4063°
2°	2.1094°	2.1094°
3°	3.1641°	2.8125°
4°	4.2188°	4.2188°
5°	4.9219°	4.9219°
10°	9.8438°	9.8438°
15°	15.1172°	14.7656°
20°	20.0391°	20.3906°
25°	24.9609°	25.3125°
30°	29.8828°	30.2344°
35°	35.1563°	35.1563°
40°	40.0781°	40.0781°
45°	45°	45°

Table 5.2: Scaling parameter detection precision with different sampling points on θ and ρ

Scaling parameter detection precision			
Applied scaling factor	Detected scaling factor with 1024×1024	Detected scaling factor with 512×512	Detected scaling factor with 512×1024
0.8	0.7985	0.8036	0.7999
0.85	0.8515	0.8512	0.8511
0.9	0.9022	0.9016	0.9055
0.95	0.9499	0.9550	0.9516
1.05	1.0528	1.0471	1.0509
1.1	1.1013	1.0964	1.1043
1.15	1.1520	1.1481	1.1462
1.20	1.1973	1.2022	1.2045

5.2.3 Calculation of translation parameters

Because the magnitude of the Fourier transform of a frame is independent of the translational parameters, we have to calculate the translational parameters in the spatial

domain instead of the Fourier transform domain. We need to cut another block in the spatial domain as $Template2$, as shown in Fig. 5.2 and compute the cross-correlation between this template and the watermarked image having undergone RST attacks by using the phase-only filtering method. Suppose the coordinate of the peak is (x_1, y_1), and we know the original position of the template is (x_0, y_0), so the translational parameters are:

$$\begin{cases} \Delta_x &= x_1 - x_0 \\ \Delta_y &= y_1 - y_0 \end{cases} \tag{5.4}$$

It is very easy to invert the translation transform applied to the watermarked frame. The detected translation parameter is very accurate because the detected values equal the real translational parameters.

5.3 A watermarking algorithm for MPEG video authentication

In order to demonstrate that our watermarking algorithm can be integrated into any independent video watermarking algorithm, we introduce a semi-fragile video watermarking algorithm in this section. A typical video watermarking algorithm to embed a high load into the least significant bit (LSB) of the DCT blocks, which called DCTLSB method, was proposed by Lin and Chang [108]. The method is working on each DCT block of an I-frame during MPEG-2 compression. 4 bits of messages can be embedded into each 8×8 DCT blocks. We use a test video, *mobile* with the size of 576×704, as an example. Each I-frame can embed 25344 bits (3168 bytes) information, which is a high load capability among the existed video watermarking algorithms.

The watermark embedding is conducted as follows:

1. Divide the target video into Group of Pictures (GOPs).
2. Divide the I-frame of each GOP into non-overlapped 8×8 blocks.
3. Apply DCT to each 8×8 blocks.
4. Select 7 coefficients $C_0, C_1, \cdots, C_6$ from the last 28 coefficients in zig-zag scan order of DCT coefficients, which means the watermark is embedded in middle or high frequency of each DCT block.
5. Divide each coefficients C_i by the corresponding quantization value Q_i and quantization scale factor α, and then, round to the nearest integer D_i.

$$D_i = round(\frac{C_i}{\alpha Q_i}) \tag{5.5}$$

6. Take the LSB L_i of each rounded integer D_i and exclusive-or them together to get a bit value b_e.

$$b_e = XOR[L_0, L_1, \cdots, L_6] \tag{5.6}$$

7. If b_e is not equal to the watermark bit needed to be embedded, then, modify one of the LSBs of the rounded integers in step 6.
8. Repeat above steps to embed other 3 bits in the same DCT blocks.
9. Repeat the whole steps to other DCT blocks to embed more watermark messages.
10. Conduct inverse quantization and inverse DCT to get the watermarked I-frame.

The watermark extraction is similar to watermark embedding, shown as follows:

1. Repeat first three steps in embedding process.

2. Select the same coefficients as the embedding process and divide by the corresponding quantization value and quantization scale factor, and round to the nearest integer.

3. Take the LSB of each rounded integer and exclusive-or them together to get a bit value which is the extracted watermark value.

4. Continue above steps until all the watermark values are extracted.

5.4 Embedding location

The integration of our LPMPOF algorithm with the DCTLSB algorithm will carry one RST robust watermark and one high-capacity watermark. We need to select carefully the location to embed these two watermark patterns.

5.4.1 Embedding in different frames

With an assumption that the whole video will undergo the same distortions, we can embed different watermark pattern to different frames and two watermark patterns will not effect each other. We call it independent watermark embedding. In our case, both LPMPOF algorithm and DCTLSB algorithm need to work on I-frame of GOPs. We can choose to apply the two algorithms to two different GOPs. Both embedding and extraction for two algorithms are independent. The RST parameters detected by the LPMPOF algorithm rectify the whole video and the DCTLSB algorithm can be applied to carry watermark message.

5.4.2 Embedding in same frame

The other possibility is to embed rectification watermark and load carrier watermark into the same I-frame of GOPs. The benefit of this embedding method is that the RST parameter detection does not rely on other frames which could be lost or distorted in different way. The drawback could be that the two watermark patterns effect by each other. Therefore, the embedding location is very important.

The ideal way to embed two patterns without effecting each other is orthogonal embedding. However, the two algorithms are using different domains for watermark embedding. The LPMPOF algorithm uses the LPM domain of the magnitude of the Fourier transform of the original frame and the DCTLSB algorithm uses the DCT domain of the original frame. These two domains are not orthogonal. Therefore, we have to find a way to reduce the mutual interference as much as possible.

The DCTLSB algorithm works on the DCT blocks of Y (luminance) components, which follows the MPEG compression procedures. Therefore, the LPMPOF algorithm will not be applied in the same component to avoid interference. The two chrominance components, U and V, are not good choice either because of the loss they will undergo during video compression. We have to consider R,G or B component for the LPMPOF algorithm. Equ. (5.7) is the equation for color space conversion from the RGB model to YUV model. B component has the least contribution to Y component. As a result, we choose B component to embed watermark for the LPMPOF algorithm whose purpose is RST parameters detection.

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{5.7}$$

5.4.3 Embedding only one watermark

In above two sections, we explain two different embedding methods to embed two different watermarking patterns. However, the watermark pattern for the LPMPOF algorithm is not necessary if this algorithm is only used for RST parameters detection. A template cut from LPM domain of original frame is enough to detect the watermark location and to obtain the rotation and scaling parameters. Translation parameters are calculated with the need of a template in the spatial domain. In this case, the LPMPOF algorithm has no interference with DCTLSB algorithm. Therefore, the detection results are same as the first embedding method, shown in Section 5.4.1.

5.5 Experimental results

We use the same test videos as in Chapter 4, shown in Fig. 4.8.

5.5.1 Watermark for DCTLSB algorithm

In our implementation, we embed a logo of University of Ottawa as the watermark message for the DCTLSB algorithm. To simplify the embedding, we choose a binary logo of size of 64×64, as shown in Fig. 5.3. In Section 5.4, we have discussed embedding locations for different scenarios. In the following, we will give the experimental results for them.

5.5.2 Embedding in different I-frames

In this embedding scenario, the two watermarking algorithms are independently working on different I-frame. They will not effect each other. All the experimental results

Figure 5.3: Logo of University of Ottawa of size of 64×64.

for the LPMPOF algorithm has been illustrated in Chapter 4. We will not repeat here. For the DCTLSB algorithm, we will only show the bit error rate after RST distortions. Note that the DCTLSB algorithm would fail absolutely if the LPMPOF algorithm was not integrated. Table 5.3 shows the bit error rate of the extracted watermark pattern compared with the original logo under RST distortions for test video *mobile* with different quantization scale factors. Q represents quantization scale factor α. Embed-2 represents embedding in two different I-frames while Embed-1 means embedding in the same I-frame. T means translation attacks. S means scaling distortion. R means rotation distortion. Table 5.4 shows the bit error rates for all other test videos with quantization scale factor of $Q = 1$.

During the watermark embedding, we choose $\alpha = 1$. During the watermark extraction, we test four different quantization scale factors, $0.7, 0.8, 0.9$ and 1. When the chosen quantization scale factor for the watermark extraction is same as the one in the watermark embedding, the bit error rate is zero, which means all the bits were extracted correctly. The closer the quantization scale factor is used in the watermark extraction to the one used in the watermark embedding, the smaller the bit error rate is after the watermark extraction. The following discusses the performance against RST distortions.

- Translation: We circularly shift the frame in the target video 10 pixels to the right. With different quantization scale factors, the highest bit error rate is around 3%, which is still acceptable.

- Scaling: The bit error rates for different scaling ratios are high. It is because the precision of scaling parameter and interpolation are not good enough to rectify the watermarked frame back to the original size and the DCTLSB algorithm is very sensitive to scaling since it works on the least significant bits. It would be our future work to improve it.

- Rotation: We apply rotation with different angles to the target video. With the small rotation angles, less than 5°, the bit error rates are acceptable.

Table 5.3: Bit error rates for test video *mobile*

Q	1		0.9		0.8		0.7	
	Embed-1	Embed-2	Embed-1	Embed-2	Embed-1	Embed-2	Embed-1	Embed-2
T	0	0	0.0392	0.0396	0.0344	0.0384	0.0332	0.0330
S 80%	0.3420	0.3440	0.3809	0.3840	0.3906	0.3918	0.3928	0.3932
85%	0.3012	0.3023	0.3334	0.3387	0.3454	0.3476	0.3477	0.3576
90%	0.2433	0.2434	0.2845	0.2839	0.3097	0.3120	0.3432	0.3443
95%	0.1555	0.1677	0.1698	0.1756	0.1876	0.1977	0.2054	0.2155
105%	0.1435	0.1466	0.1576	0.1587	0.1622	0.1699	0.1706	0.1777
110%	0.2347	0.2350	0.2564	0.2570	0.2894	0.2890	0.3020	0.3025
115%	0.2556	0.2587	0.2600	0.2607	0.2744	0.2797	0.2895	0.2900
120%	0.2890	0.2896	0.3095	0.3096	0.3185	0.3196	0.3755	0.3780
R 1°	0.0440	0.0448	0.0468	0.0472	0.0556	0.0548	0.0760	0.0768
2°	0.0880	0.0868	0.0932	0.0932	0.1116	0.1116	0.1272	0.1276
3°	0.1024	0.1120	0.1060	0.1060	0.1248	0.1248	0.1456	0.1464
4°	0.1380	0.1476	0.1536	0.1548	0.1692	0.1700	0.1992	0.1996
5°	0.1628	0.1664	0.1700	0.1708	0.1872	0.1864	0.2176	0.2160
10°	0.2756	0.2852	0.2920	0.2980	0.3268	0.3264	0.3672	0.3680
15°	0.3252	0.3340	0.3408	0.3428	0.3576	0.3612	0.3908	0.3928
20°	0.3532	0.3536	0.3724	0.3740	0.3900	0.3892	0.4156	0.4192
25°	0.3560	0.3576	0.3820	0.3840	0.4136	0.4120	0.4252	0.4208
30°	0.3684	0.3688	0.3828	0.3856	0.3952	0.4004	0.4272	0.4284
35°	0.3732	0.3808	0.3784	0.3788	0.3928	0.3924	0.4244	0.4260
40°	0.3880	0.3956	0.4004	0.4016	0.4124	0.4128	0.4296	0.4304
45°	0.4036	0.4144	0.4180	0.4212	0.4224	0.4232	0.4236	0.4280

Table 5.4: Bit error rates for other four videos with $Q = 1$

Test videos	Football		Foreman		Garden		Table Tennis	
	Embed-1	Embed-2	Embed-1	Embed-2	Embed-1	Embed-2	Embed-1	Embed-2
T	0	0	0	0	0	0	0	0
S 80%	0.3034	0.3065	0.3365	0.3433	0.3056	0.3121	0.3232	0.3376
85%	0.2987	0.3022	0.3211	0.3276	0.2865	0.2899	0.3043	0.3122
90%	0.2343	0.2387	0.2743	0.2812	0.2434	0.2543	0.2435	0.2544
95%	0.1432	0.1543	0.1687	0.1708	0.1376	0.1422	0.1543	0.1654
105%	0.1324	0.1433	0.1511	0.1599	0.1254	0.1344	0.1502	0.1587
110%	0.2265	0.2333	0.2454	0.2533	0.2355	0.2397	0.2345	0.2432
115%	0.2487	0.2566	0.2654	0.2698	0.2564	0.2614	0.2654	0.2765
120%	0.2898	0.2987	0.2865	0.2908	0.2998	0.3042	0.2943	0.3021
R 1°	0.0321	0.0387	0.0301	0.0343	0.0312	0.0345	0.0299	0.0321
2°	0.0798	0.0856	0.0765	0.0798	0.0698	0.0732	0.0587	0.0654
3°	0.1123	0.1276	0.1044	0.1076	0.1187	0.1198	0.0976	0.0987
4°	0.1400	0.1499	0.1387	0.1465	0.1454	0.1532	0.1343	0.1355
5°	0.1633	0.1723	0.1545	0.1587	0.1543	0.1565	0.1488	0.1522
10°	0.2678	0.2765	0.2977	0.2981	0.2587	0.2621	0.2508	0.2633
15°	0.3307	0.3454	0.3387	0.3388	0.3276	0.3329	0.3045	0.3191
20°	0.3654	0.3697	0.3545	0.3622	0.3567	0.3598	0.3421	0.3479
25°	0.3765	0.3787	0.3755	0.3778	0.3654	0.3719	0.3564	0.3630
30°	0.3822	0.3821	0.3828	0.3856	0.3756	0.3792	0.3765	0.3876
35°	0.3876	0.3912	0.3898	0.3900	0.3865	0.3876	0.3796	0.3788
40°	0.3921	0.3987	0.3943	0.3987	0.3876	0.3930	0.3854	0.3987
45°	0.4265	0.4377	0.4022	0.4126	0.4232	0.4344	0.4122	0.4212

5.5.3 Embedding in same I-frame

In this embedding scenario, two watermarking algorithms work on the same I-frame. The LPMPOF algorithm works on B-components while the DCTLSB algorithm works on Y-components. According to the equation for the color space conversion (Equ. (5.7)), the LPMPOF algorithm should have little affect on the bit error rate of the DTLSB method. As shown in Table 5.3, the bit error rates are only slightly higher than the independent embedding, which shows that these algorithms can work together very well even when they are applied on the same I-frame.

5.5.4 Improving bit error rates by error control coding

It is known that error control coding can improve bit error rates for the information delivered from a source to a destination. In this section, we introduce a convolutional coding - Trellis encoding and Viterbi decoding as a maximum likelihood decoding for convolutional codes [109] to do error correction for our watermarking system.

- Trellis encoding

  Trellis encoder is a finite state machine [109]. The trellis structure, we used here, is shown in Fig. 5.4. We define 8 states in our trellis structure, which are represented by $\{A0, B0, C0, D0, E0, F0, G0, H0\}$ in the left side of the structure. There are two types of input path, the solid line and the dashed line. When input is 0, it goes through the solid line, while 1 goes through the dashed line. There are 16 output symbols, which are $0000, 0001, 0010, 0011, \cdots, 1111$. For each path, there is a reference associated to it, shown on the top of each path in Fig. 5.4. Fig. 5.4 shows an example to encode a 4-bit input $\{0110\}$. The output bits are $\{0000\ 0001\ 0011\ 1000\}$. Fig. 5.4 highlights the code words for input $\{0110\}$.

- Viterbi decoding

  After Trellis encoding, the input code will be spread by 4 times and converted to the code words, and then transmitted across a noisy channel. The Viterbi decoder, as a convolutional decoder, receives the distorted code words and generates the decoded words. It uses the maximum likelihood (ML) method to select the estimated decoded words with the maximum probability through the Trellis structure. In hard-decision decoding, the most likely path $P(state1, state2)$ with the maximum inner product between the received code r and the reference marks $m(state1, state2)$ along the path is chosen, described as follows:

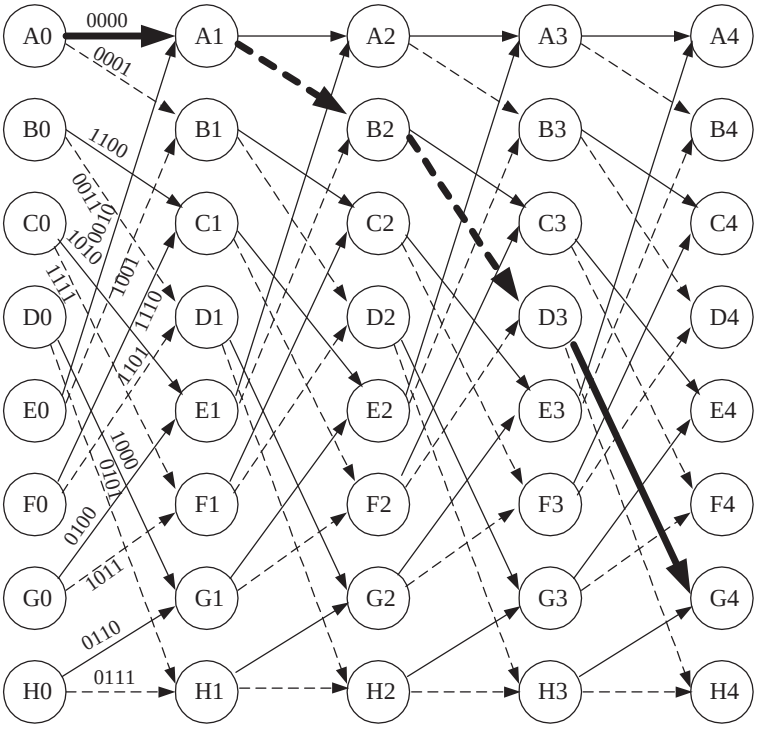


Figure 5.4: Trellis structure with 8 states.

$$P(state1, state2) = MAX(r \cdot m(state1, state2)) \tag{5.8}$$

$MAX(\cdot)$ means the path with the biggest inner product among all the pathes is chosen. "$\cdot$" is the inner product. The whole path from the starting state to the end state is decided with all $P(state1, state2)$ along the Trellis structures.

We apply the Trellis encoding to the logo at the beginning of watermark embedding. Then, we decode the retrieved watermark by the Viterbi decoder. Table 5.5 shows the bit error rates with the ECC enhancement for video *mobile* with different quantization scale factors. Table 5.6 shows the bit error rates with ECC enhancement for other test videos with quantization scale factor of $Q = 1$. All the error rates are improved by

the ECC in different percentages. For translation and rotation, the bit error rates are improved by at least 50%. For scaling, the improvement is smaller because the DCTLSB algorithm is more sensitive to scaling distortions and the implementation precision for scaling is less than other two distortions.

The experimental results about bit error rates for RST show that the LPMPOF algorithm does help the semi-fragile watermarking algorithm (DCTLSB) to survive RST distortions to a certain degree. In the original algorithm [108], the threshold is set to be 85%, which means when more than 85% of the watermark bits are correctly detected, the watermarked and the original frames are considered to match. Using the same threshold, with the help of the LPMPOF algorithm, the DCTLSB algorithm is robust to translation, rotation up to 10° and scaling up to 95% and 105%.

Table 5.5: Bit error rates with ECC for video *mobile*

Q factor	1		0.9		0.8		0.7	
	Embed-1	Embed-2	Embed-1	Embed-2	Embed-1	Embed-2	Embed-1	Embed-2
T	0	0	0	0	0	0	0.0032	0.0038
S 80%	0.2240	0.2396	0.2305	0.2465	0.2475	0.2401	0.2876	0.2922
85%	0.1940	0.2096	0.2005	0.2065	0.2275	0.2301	0.2676	0.2722
90%	0.1234	0.1343	0.1365	0.1454	0.1422	0.1497	0.1543	0.1600
95%	0.0834	0.0943	0.0965	0.0954	0.1022	0.1097	0.1243	0.1300
105%	0.0630	0.0743	0.0743	0.0867	0.0809	0.0843	0.0933	0.0987
to 110%	0.1430	0.1543	0.1443	0.1567	0.1609	0.1843	0.1933	0.1987
to 115%	0.2230	0.2243	0.2343	0.2367	0.2409	0.2443	0.2633	0.2687
to 120%	0.2590	0.2676	0.2687	0.2689	0.2698	0.2732	0.2875	0.2954
R 1°	0.0034	0.0036	0.0073	0.0082	0.0061	0.0060	0.0129	0.0145
2°	0.0112	0.0122	0.0125	0.0164	0.0326	0.0328	0.0269	0.0328
3°	0.0146	0.0150	0.0168	0.0177	0.0232	0.0244	0.0376	0.0378
4°	0.0220	0.0226	0.0273	0.0303	0.0366	0.0406	0.0635	0.0650
5°	0.0303	0.0311	0.0369	0.0380	0.0508	0.0510	0.0896	0.0900
10°	0.0669	0.0680	0.0840	0.0850	0.1289	0.1280	0.1314	0.1410
15°	0.1206	0.1240	0.1414	0.1455	0.1890	0.1888	0.2234	0.2321
20°	0.1680	0.1689	0.1892	0.1898	0.2371	0.2408	0.2654	0.2645
25°	0.1641	0.1655	0.1829	0.1855	0.2219	0.2231	0.2825	0.2890
30°	0.1624	0.1633	0.1887	0.1876	0.2273	0.2343	0.2778	0.2876
35°	0.1812	0.1823	0.2017	0.2212	0.2346	0.2344	0.2729	0.2744
40°	0.2087	0.2342	0.2261	0.2343	0.2466	0.2457	0.2905	0.3087
45°	0.2415	0.2455	0.2546	0.2576	0.2810	0.2865	0.3042	0.3140

Table 5.6: Bit error rates with ECC for other four videos with $Q = 1$

Test videos	Football		Foreman		Garden		Table Tennis	
	Embed-1	Embed-2	Embed-1	Embed-2	Embed-1	Embed-2	Embed-1	Embed-2
T	0	0	0	0	0	0	0	0
S 80%	0.2121	0.2154	0.2235	0.2325	0.2320	0.2345	0.2021	0.2122
85%	0.1987	0.2010	0.2009	0.2034	0.2132	0.2176	0.1921	0.1966
90%	0.1134	0.1232	0.1321	0.1366	0.1332	0.1430	0.1187	0.1200
95%	0.0754	0.0866	0.0832	0.0865	0.0965	0.0992	0.0765	0.0897
105%	0.0674	0.0677	0.0732	0.0768	0.0865	0.0887	0.0622	0.0676
to 110%	0.1454	0.1445	0.1234	0.1324	0.1543	0.1655	0.1343	0.1435
to 115%	0.2123	0.2254	0.1973	0.2067	0.2186	0.2233	0.1876	0.1987
to 120%	0.2656	0.2765	0.2454	0.2569	0.2323	0.2394	0.2076	0.2233
R 1°	0.0032	0.0034	0.0034	0.0033	0.0030	0.0031	0.0030	0.0033
2°	0.0122	0.0130	0.0112	0.0122	0.0109	0.0128	0.0100	0.0123
3°	0.0143	0.0154	0.0135	0.0139	0.0143	0.0154	0.0123	0.0124
4°	0.0212	0.0234	0.0232	0.0234	0.0209	0.0212	0.0189	0.0198
5°	0.0323	0.0346	0.0321	0.0324	0.0365	0.0387	0.0265	0.0256
10°	0.0567	0.0592	0.0654	0.0690	0.0633	0.0640	0.0543	0.0554
15°	0.1298	0.1265	0.1270	0.1324	0.1260	0.1340	0.1123	0.1210
20°	0.1567	0.1675	0.1502	0.1694	0.1765	0.1830	0.1540	0.1592
25°	0.1776	0.1768	0.1679	0.1765	0.1798	0.1876	0.1765	0.1798
30°	0.1789	0.1876	0.1701	0.1893	0.1820	0.1879	0.1862	0.1892
35°	0.1890	0.1977	0.1927	0.2098	0.1987	0.2228	0.2087	0.2387
40°	0.2321	0.2450	0.2321	0.2309	0.2345	0.2470	0.2230	0.2392
45°	0.2459	0.2576	0.2560	0.2693	0.2765	0.2904	0.2459	0.2545

5.6 Conclusion

In this chapter, we demonstrated that our LPMPOF algorithm can provide the DCTLSB algorithm with robustness to RST distortions. Considering the fragility of DCTLSB algorithm, the LPMPOF algorithm should work well with most existing video watermarking algorithms.

Chapter 6

A novel video watermarking algorithm based on 1D DFT and 1D projection

In this chapter, we propose a novel video watermarking algorithm robust against geometric attacks and H.264 compression.

First, we will introduce two enabling techniques - the 1D DFT in temporal direction and the 1D projection 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 1D projection. Gradient method is also adopted to improve the watermark detection.

Figure 6.1: Three consecutive frames.

6.1 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 with 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 [110]:

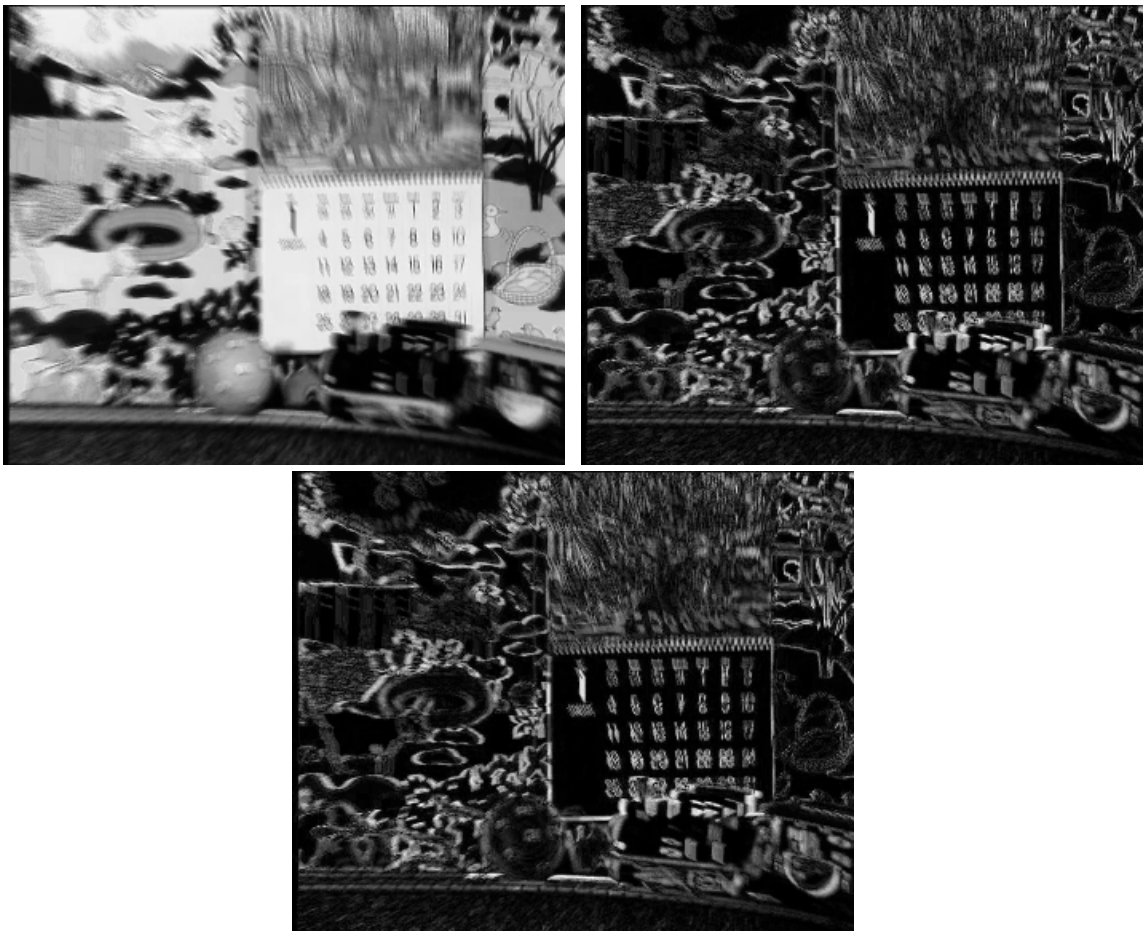
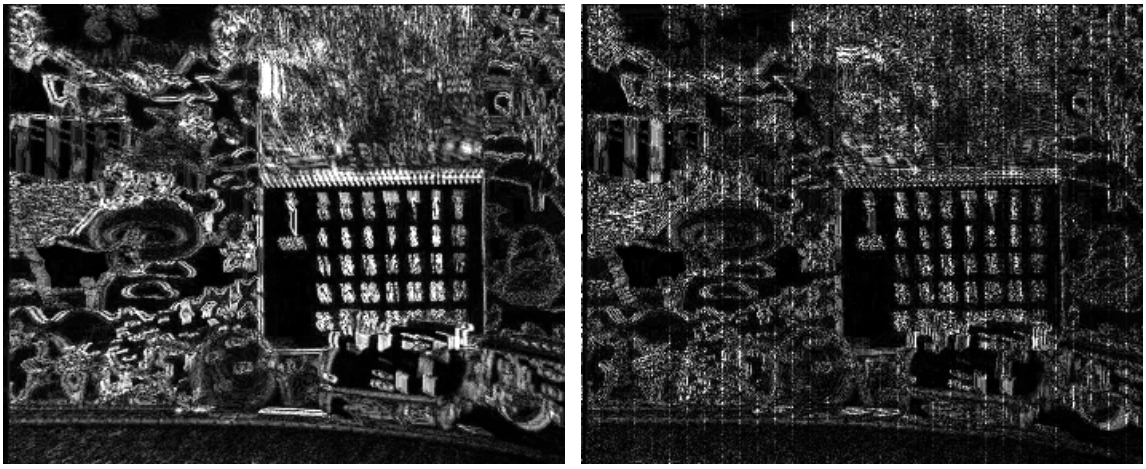


Figure 6.2: The 1D DFT along temporal direction of the three consecutive frames in Fig. 6.1.

$$F(x, y, \tau) = \sum_{t=0}^{T-1} f(x, y, t) e^{-j2\pi(t\tau/T)} \tag{6.1}$$

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

$$f(x, y, t) = \frac{1}{T} \sum_{\tau=0}^{T-1} F(x, y, \tau) e^{j2\pi(t\tau/T)} \tag{6.2}$$



(a) Selected original frame from temporal frequency domain.

(b) Watermarked version of the frame in (a).

Figure 6.3: Original and watermarked frame in temporal frequency domain.

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

$$F(x, y, \tau) = |F(x, y, \tau)|e^{j\phi(u,v,\tau)} \tag{6.3}$$

$$|F(x, y, \tau)|^2 = Re^2(F(x, y, \tau)) + Im^2(F(x, y, \tau)) \tag{6.4}$$

$$\phi(x, y, \tau) = tan^{-1}\Big[\frac{Im(F(x, y, \tau))}{Re(F(x, y, \tau))}\Big] \tag{6.5}$$

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

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

Fig. 6.1 and Fig. 6.2 give an example of three consecutive frames and the frequency magnitude after 1D DFT along temporal direction, respectively. In the temporal frequency frames, all the spatial information remains and is partially cumulate in each frame. The temporal frequency information is also 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.

6.2 1D projection

As mentioned in Section 2.3 in Chapter 2, the Radon transform generates with a collection of projections along various direction [31]. In our case, 1D projection along vertical direction is used during Radon transform with an angle γ equal to 0, to transfer a frame into an one dimensional array, as expressed as follows:

$$G(x,\tau) = \sum_{y=0}^{N-1} F(x,y,\tau) \tag{6.6}$$

where, $F(x,y,\tau)$ is the τ^{th} frame in the temporal frequency domain, with a size of $M \times N$. $G(x,\tau)$ is the 1D projection of $F(x,y,\tau)$ along the vertical direction (y direction). $G(x,\tau)$ is an one dimensional array with a length of M.

6.3 Implementation strategies

In this section, we will explain some important implementation strategies for our video watermarking algorithm in order to be robust against RST attacks and H.264 compression.

6.3.1 Video compression and watermark location selection

Video frames are considered as three dimensional information, which includes two dimensional spatial information and one dimensional 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 spatial 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 by 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 the high temporal frequency.

In general, all block based video compressions, MPEG-2, MPEG-4 and H.264, follow the same rule, which is removing high spatial frequencies and low temporal frequencies as much as possible. H.264 has the most efficient methods to fulfill this requirement among three of mentioned video compression standards. From the features of H.264, we can conclude that H.264 has much more efficiency on inter-picture prediction than

MPEG-2 and MPEG-4. With the use of H.264, more temporal redundancies are removed and up to 50% of bit rate could be saved [40].

These analysis give us some ideas for perfect watermark location - some places that have low or middle spatial frequency and high temporal frequency are the safest locations for watermark embedding against video compression. High temporal frequency locations are more important for watermark embedding. As a result, we introduce the 1D DFT along temporal direction which could give us a clear way to find the high temporal frequencies in our novel video watermarking algorithm, which will be explained in Section 6.3.4.

6.3.2 Vertical line embedding for robustness to RST attacks

As we mentioned in Section 6.3.1, video compression is one of most important attacks to a video watermarking algorithm. Similar to compression, noise addition, filtering, and frame loss 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 [1][5][68][29] 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.

6.3.3 Embedding method optimization based on fidelity evaluation

6.3.3.1 Fidelity evaluation method

First, we introduce two video quality evaluation methods. Peak Signal to Noise Ratio (PSNR) is a well known objective measurement method. The structural Similarity based quality measurement (SSIM) is recently created by Wang et. al. [7].

The PSNR is defined as:

$$PSNR = 10log_{10}\frac{P_{max}}{MSE} \tag{6.7}$$

where, P_{max} is the maximum intensity value of the video signal. For 8-bit image, $P_{max} = 255$. MSE is the mean squared error. However, it is not very well matched to perceived visual quality [111].

SSIM - Structural Similarity based quality measurement is a more direct measurement way to compare the structures of the reference and the degraded signals [7]. It is closer to human visual system (HVS). The similarity measurement is combined with three components, Luminance comparison $l(x, y)$, Contrast comparison $c(x, y)$ and

Structure comparison $s(x, y)$:

$$SSIM(x, y) = f(l(x, y), c(x, y), s(x, y)) \tag{6.8}$$

where, $f(.)$ is the combination function. As a result, the similarity metric between two non-negative image signals x and y of common size $N \times N$ is:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{6.9}$$

where, μ_x and μ_y are the mean intensity of x and y, respectively:

$$\begin{cases} \mu_x &= \frac{1}{N}\sum_{i=1}^{N} x_i \\ \mu_y &= \frac{1}{N}\sum_{i=1}^{N} y_i \end{cases} \tag{6.10}$$

σ_x^2 and σ_y^2 are the variance of x and y, respectively:

$$\begin{cases} \sigma_x^2 &= \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2 \\ \sigma_y^2 &= \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \mu_y)^2 \end{cases} \tag{6.11}$$

σ_{xy} is the covariance of x and y:

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y) \tag{6.12}$$

c_1 and c_2 are two variables to stabilize the division with weak denominator:

$$\begin{cases} c_1 = (K_1 L)^2 \\ c_2 = (K_2 L)^2 \end{cases} \tag{6.13}$$

where, $K_1 \ll 1$ and $K_2 \ll 1$, and L is the dynamic range of the pixel values (255 for 8-bit image). The maximum index of SSIM is 1, which means two blocks are exactly the same.

6.3.3.2 Optimization of watermark embedding

We embed watermark pattern in each vertical line of the target frames in the 1D DFT domain of the original video clip. Each pixel of the embedding vertical lines of the selected frame is replaced by the average watermark value. We call it average embedding method as follows:

$$M(u, v, \tau) = \begin{cases} w(u) \cdot \frac{\alpha}{N} & w(x) \neq 0 \\ F(x, y, \tau) & w(x) = 0 \end{cases} \tag{6.14}$$

where, $M(x, y, \tau)$ represents the pixel value after watermarking; $F(x, y, \tau)$ is the original value before watermarking in the temporal frequency domain; $w(x)$ is the watermark value at the horizontal position u; N is the vertical size of the target frame; and α is the watermark embedding strength.

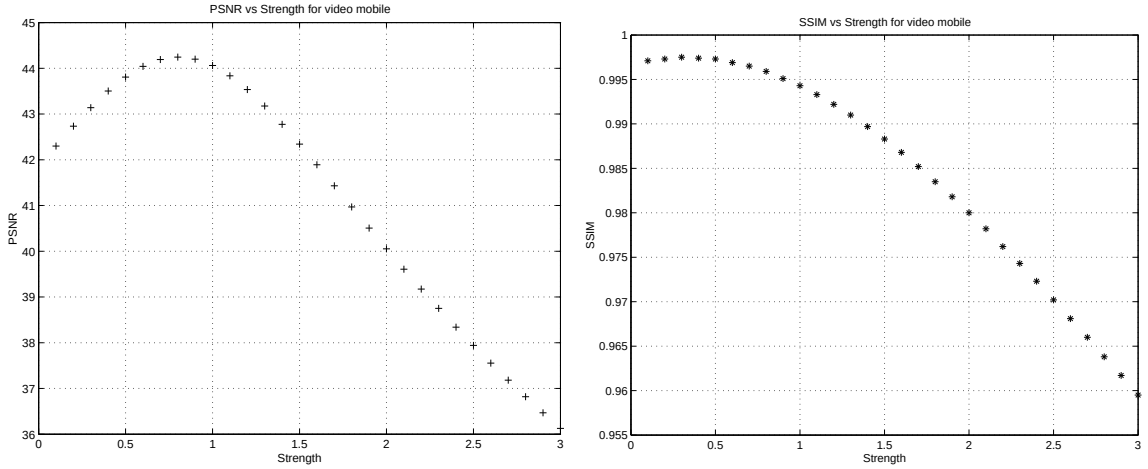
By carefully selecting the embedding strength, we can make the balance of the fidelity of watermarked video and robustness to various attacks according to the objective video quality evaluation method PSNR. However, SSIM does not give the homogeneous results as PSNR. Fig. 6.4 shows that the highest points for PSNR and SSIM are not

matching with the same embedding strength. The best quality indication based on PSNR is with the embedding strength of 0.8, while it is 0.3 based on SSIM. In the other words, the best quality indication for PSNR and SSIM has the big difference. Therefore, we need to optimize our embedding method to get the matching point that the objective and subject method has the similar evaluation results.

Base on the idea above, we improve our embedding by using proportional embedding method, as follows:

$$M(x,y,\tau) = \begin{cases} w(x)\cdot\frac{\alpha}{G(x,\tau)}\cdot F(x,y,\tau) & w(x) \neq 0 \\ F(x,y,\tau) & w(x) = 0 \end{cases} \tag{6.15}$$

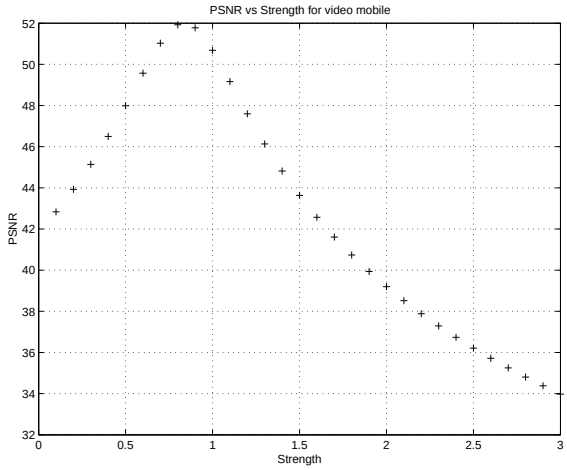
where, $G(x,\tau)$ is the 1D projection of $F(x,y,\tau)$ along the vertical direction (v direction) and is an one dimensional array with a length of M; M is the horizontal size of the target video frame. Fig. 6.5 shows that PSNR and SSIM agree to each other for the best embedding strength under proportional embedding method.



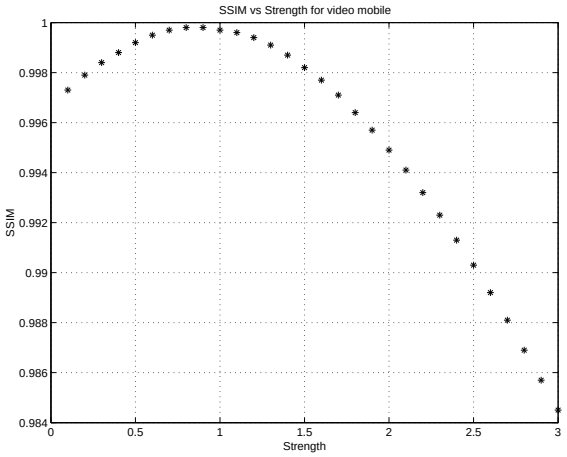
(a) PSNR vs. embedding strength with average embedding method.

(b) SSIM vs. embedding strength with average embedding method.

Figure 6.4: PSNR vs. SSIM with average embedding method.

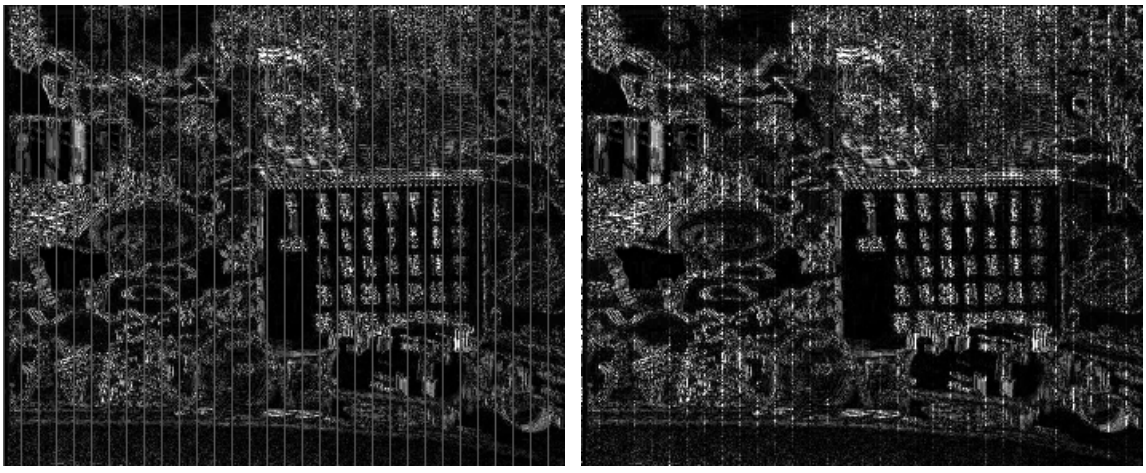


(a) PSNR vs. embedding strength with proportional embedding method.



(b) SSIM vs. embedding strength with proportional embedding method.

Figure 6.5: PSNR vs. SSIM with proportional embedding method.



(a) Average embedding method.

(b) Proportional embedding method.

Figure 6.6: Average and proportional embedding methods.

Fig. 6.6 shows the watermarked frame after two different embedding methods. The proportional embedding method does not change the frequency distribution in the target frame significantly, shown in (b). It helps to keep the fidelity of the watermarked video better than average embedding method, shown in (a).

6.3.4 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 frames 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. $f(x, y, t), t \in (0, \cdots, T-1)$ are all frames in the spatial domain in one GOP. $F(x, y, \tau), \tau \in (0, \cdots, T-1)$ are all frames in temporal frequency domain in the same GOP. $F(x, y, 0)$ represents the DC frame in the temporal frequency domain. $G(x, \tau)$ 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 $F(x, y, 1)$ and $F(x, y, T-1)$:

$$f(x, y, t)' = f(x, y, t)(1 + \frac{\alpha}{T \cdot G(x, \tau)}(w \cdot e^{j2\pi \frac{1}{T} t} + w \cdot e^{j2\pi \frac{T-1}{T} t})) \tag{6.16}$$

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

$$f(x, y, t)' = f(x, y, t)(1 + \frac{\alpha}{T \cdot G(x, \tau)}(w \cdot e^{j2\pi \frac{2}{T} t} + w \cdot e^{j2\pi \frac{T-2}{T} t})) \tag{6.17}$$

- $\cdots$

- If embed watermark in frame pair $F(x, y, \frac{T-1}{2})$ and $F(x, y, \frac{T+1}{2})$ and when T is odd:

$$f(x, y, t)' = f(x, y, t)(1 + \frac{\alpha}{T \cdot G(x, \tau)}(w \cdot e^{j2\pi \frac{T-1}{2T} t} + w \cdot e^{j2\pi \frac{T+1}{2T} t})) \tag{6.18}$$

  or, if embed watermark in frame $F(x, y, \frac{T}{2} + 1)$ and when T is even:

$$f(x, y, t)' = f(x, y, t)(1 + \frac{\alpha}{T \cdot G(x, \tau)}(w \cdot e^{j2\pi \frac{T+2}{2T} t})) \tag{6.19}$$

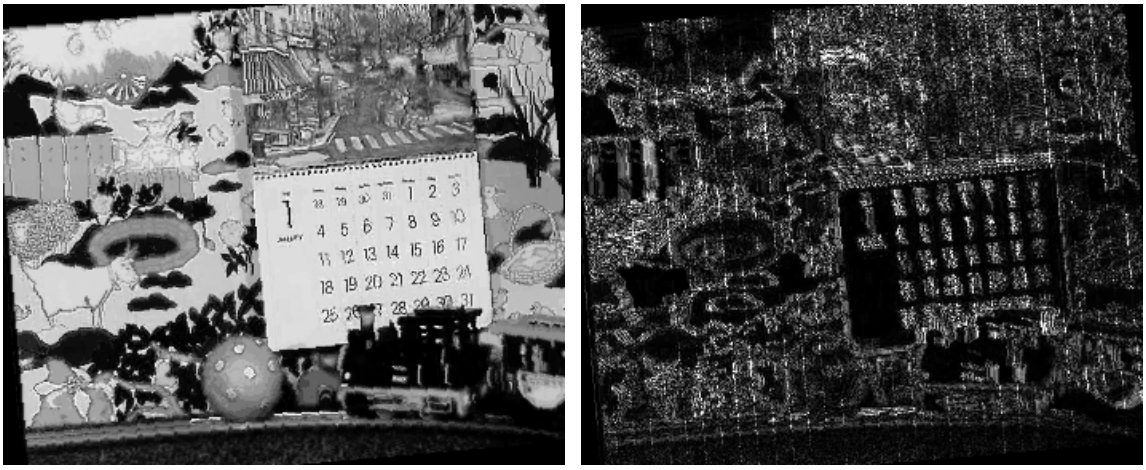
where, $f(x, y, t)$ and $f(x, y, t)'$ are respectively the original and watermarked frame in spatial domain, and w is the watermark pattern.

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:

$$f(x,y,0)' = f(x,y,0)(1 + \frac{a \cdot \alpha \cdot w}{T \cdot G(x,\tau)}) \tag{6.20}$$

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 effected but the modification will be smaller.



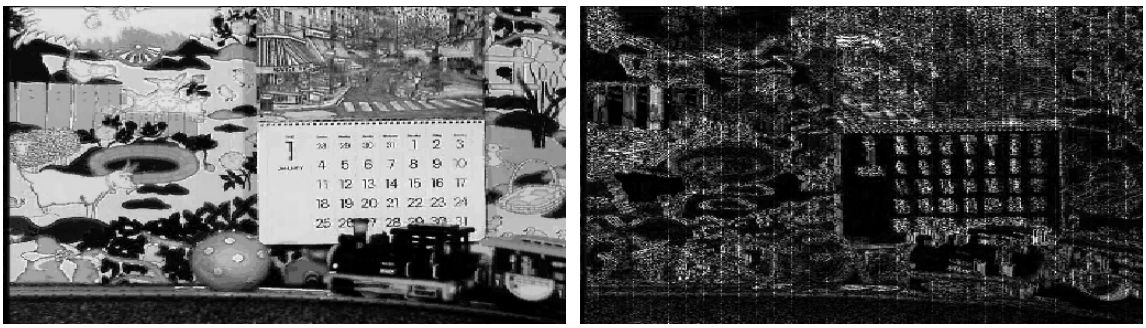
(a) A video frame in spatial domain rotated 5°.

(b) Watermarked video frame in 1D DFT domain corresponding to (a).

Figure 6.7: Original and watermarked frame in a rotated video.

6.3.5 Robustness to RST attacks

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. 6.7 (a) shows an original frame from a video slightly rotated with 5°. Fig. 6.7 (b) is the watermarked frame in the 1D DFT domain of a rotated video corresponding to the frame showed in Fig. 6.7 (a). It is reasonable



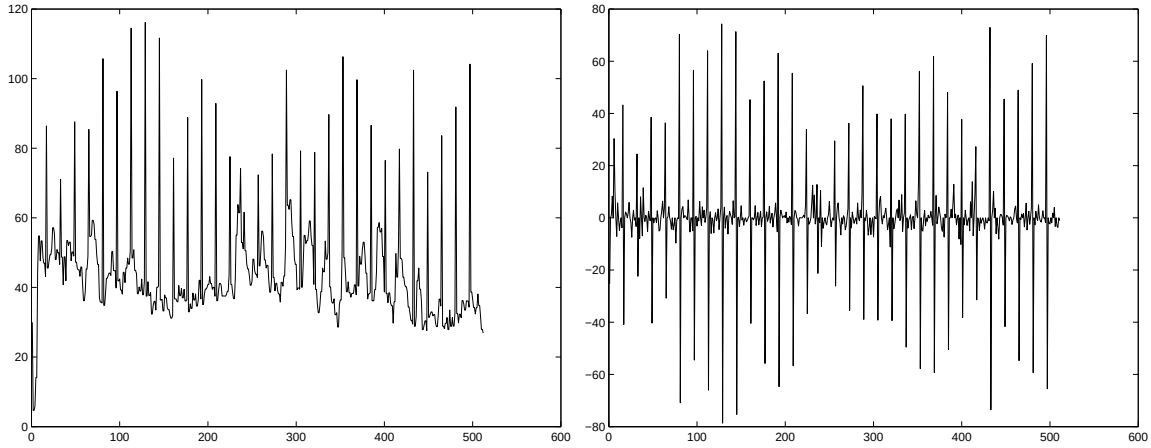
(a) A video frame in spatial domain with aspect ratio changed to 16:9 from 11:9.

(b) Watermarked video frame in 1D DFT domain corresponding to (a).

Figure 6.8: Original and watermarked frame in an aspect-ratio-changed video.

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 de-synchronize the watermark. In the proposed scheme, the gradient of the Radon transform of the watermarked frame, mentioned in Section 6.5, indicates clearly the location of the watermark pattern. Fig. 6.8 (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. 6.8 (b) shows the watermarked frame extracted from the video corresponding to Fig. 6.8 (a). Fig. 6.9 (a) presents the Radon transform of the watermarked frame in Fig. 6.8 (b); while Fig. 6.9 (b) shows the gra-



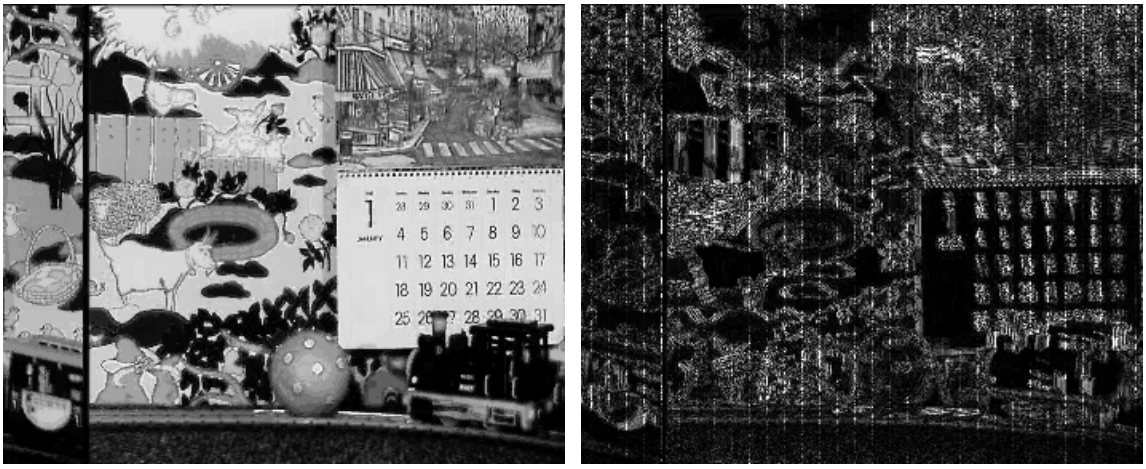
(a) The Radon transform of the watermarked frame in Fig. 6.8 (*b*).

(b) The gradient of Radon transform in Fig. 6.9 (*a*).

Figure 6.9: Radon transform of the watermarked frame after frame aspect ratio changes (a) and its gradient (b).

dient of the Radon transform in Fig. 6.9 (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. 6.10 (*a*) shows a frame from a translated video, which has been translated 50 pixels to the right. Fig. 6.10 (*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. 6.11 (a)



(a) A video frame in spatial domain translated 50 pixels to the right.

(b) Watermarked video frame in 1D DFT domain corresponding to (a).

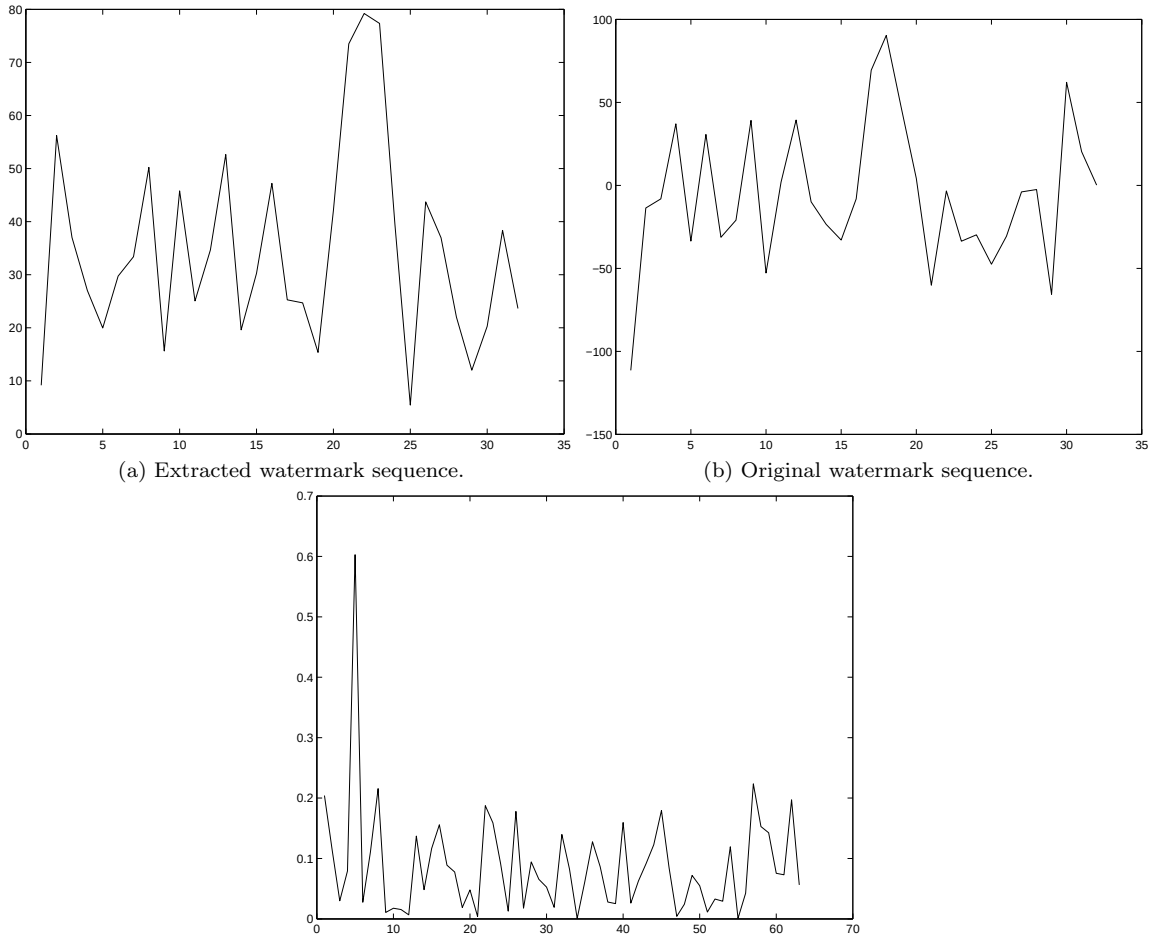
Figure 6.10: Original and watermarked frame in a translated video.

shows the retrieved watermark sequence from the Radon transform of the watermarked frame. Fig. 6.11 (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 papers [47][100], could give us a sharp peak to indicate the matching value. The filtering process is described as follows:

$$R = IFFT[F_\phi(u,v) \cdot G^*_\Phi(u,v)] \tag{6.21}$$

where

$$F_\phi(x,y) = e^{-j\Phi_F(x,y)} \tag{6.22}$$



(a) Extracted watermark sequence.

(b) Original watermark sequence.

(c) Correlation between (a) and (b) by using phase-only filtering method.

Figure 6.11: Re-synchronization of the watermark sequence for the translated video.

Fig. 6.11 (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.

6.3.6 Minimum requirements for the target video

According to video compression analysis in Section 6.3.1, 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.

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 average PSNR of all the watermarked videos have been controlled to be around 35 dB, which is considered to be of very good fidelity. Normally, we consider a good quality of a watermarked image with PSNR of 40 dB. In case of watermarked video, human eyes could not catch every details in the moving frames. The watermark is still invisible even with the lower PSNR around 35 dB for video applications. The original frame and the watermarked frame for one of our target videos are shown in Fig. 6.13.

6.4 Watermark embedding

We embed the watermark pattern into the luminance component because human visual system is more sensitive to luminance than the color and the chrominance are subsampled to the lower rate during compression.

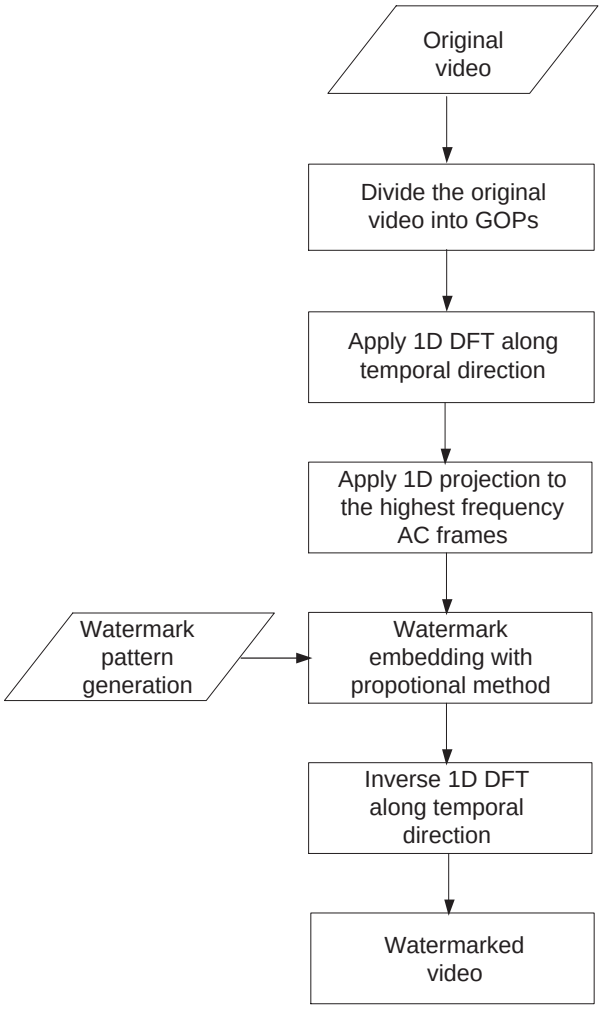
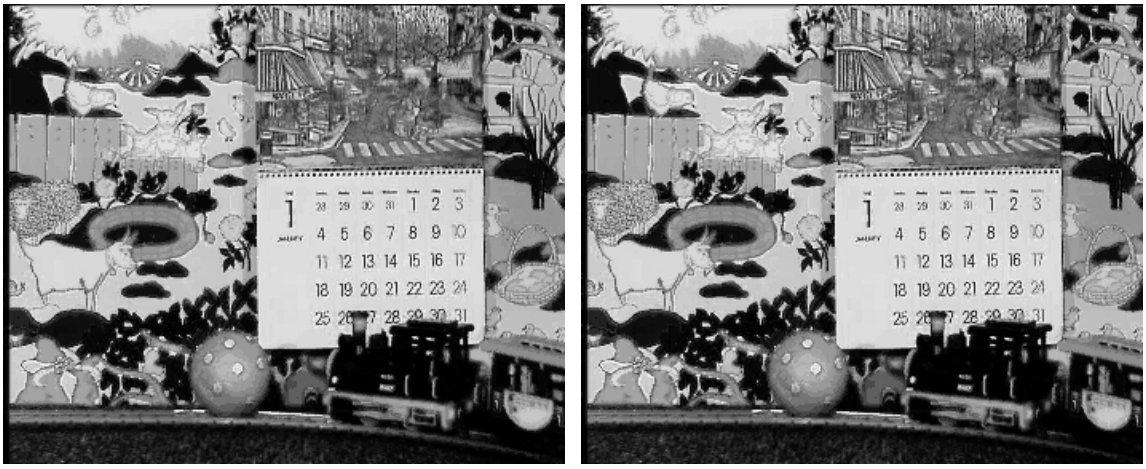


Figure 6.12: Watermark embedding procedure.



(a) First frame of video "Mobile". (b) First frame of watermarked video "Mobile".

Figure 6.13: Original and watermarked frame.

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 is 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, as shown in Fig. 6.12:

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 64 random numbers.

   Then, we spread the generated random numbers into watermark pattern w by zero-padding in the between of the generated random numbers. The length of w equals the horizontal width of target video frame.

4. Apply 1D projection to the chosen frame.

5. Embed the watermark pattern directly into the target frame (τ) by using the proportional embedding method.

   Please note we directly embed watermark in the temporal frequency domain ($F(x, y, \tau)$) instead of 1D projection domain ($G(x, \tau)$) by using Equ. (6.15). However, this embedding method produces an affect of embedding watermark in the 1D projection domain, to the watermark extraction process.

6. Finally apply the inverse 1D DFT with the modified magnitudes and the original phases of the luminance component. Then, convert this component back together with other two components 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. 6.3 shows the unwatermarked and watermarked frame 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. 6.13 shows an original frame from video "Mobile" and its corresponding watermarked frame with a PSNR of 35.7357 dB. The watermark is invisible in spatial domain although it is visible in temporal frequency domain, as shown in Fig. 6.3 and Fig. 6.13.

6.5 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, as shown in Fig. 6.14:

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.

3. Apply 1D projection to this frame (τ) to get an array $G'(u, \tau)$. Fig. 6.15 (a) displays the Radon transform of the watermarked frame in Fig. 6.3 (b). This watermarked frame has not undergone any attack. The watermark location can be detected by the sharp low or high values in the 1D projection domain.

4. Calculate the gradient of the 1D projection. The gradient of a scalar function $G'(u, \tau)$ with respect to a vector variable $u = (u_1, \cdots, u_n)$ is represented by $\nabla G'$.

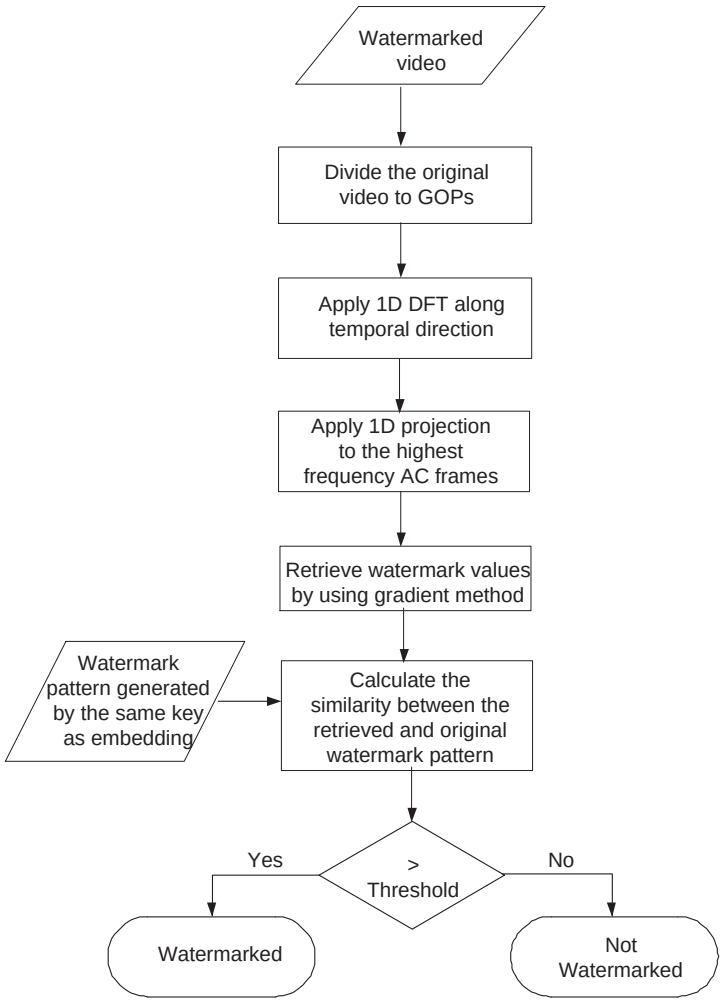
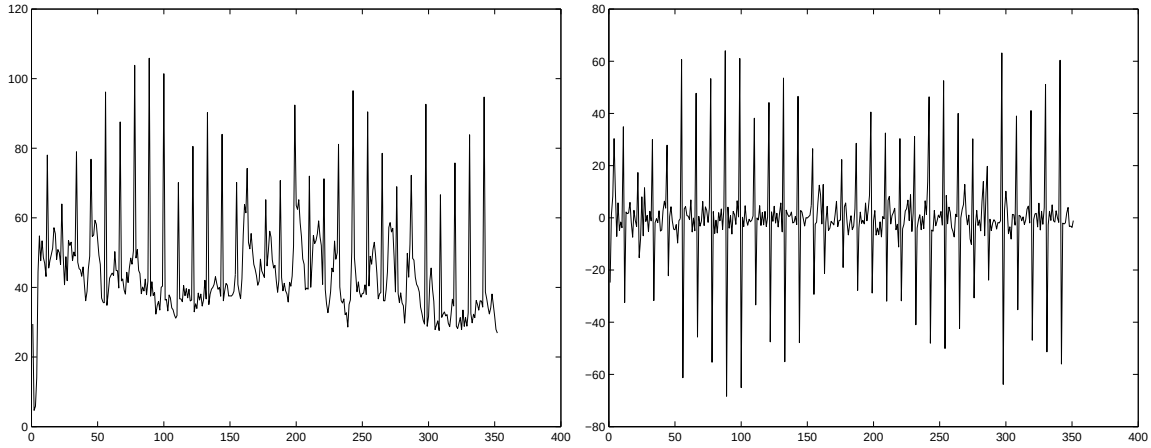


Figure 6.14: Watermark detection procedure.



(a) Radon transform of the watermarked frame in Fig. 6.3 (b).

(b) The gradient of Fig. 6.15 (a).

Figure 6.15: Radon transform of the watermarked frame in Fig. 6.3 (b) and its gradient.

It is defined to be the vector field whose components are the partial derivatives of G' as follows:

$$\nabla G' = (\frac{\partial G'}{\partial u_1}, \cdots, \frac{\partial G'}{\partial u_n}) \tag{6.23}$$

where,

$$\frac{\partial G'}{\partial u_i} \approx \frac{G'(u_{i+1}, \tau) - G'(u_i, \tau)}{u_{i+1} - u_i} \tag{6.24}$$

Fig. 6.15 (b) shows the gradient of the 1D projection in Fig. 6.15 (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 Equ. (6.23) and Equ. (6.24).

7. Calculate the normalized correlation between the gradient of the 1D projection

of the watermarked frame and the gradient of the generated watermark sequence with Equ. (6.25). 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.

$$sim = \frac{\nabla G' \times \nabla w^T}{\sqrt{(\nabla G' \times \nabla G'^T)(\nabla w \times \nabla w^T)}} \tag{6.25}$$

where, $\nabla G'$ and ∇w are, respectively, the gradient of the 1D projection of the watermarked frame and the gradient of the generated watermark sequence.

6.6 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. 4.8. We use 5 test videos, and their sizes are shown in Table 6.1. In Table 6.1, "T" and "R" respectively represent translation and rotation.

6.6.1 Fidelity

We use PSNR as an objective method to check the fidelity of the watermarked video. We compute PSNR for the luminance component 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 6.1. Human eyes could not recognize the difference between the original video and the watermarked video under these PSNR's. The average PSNR for a video would be a bit lower than the image since human eyes could not catch every motion details.

Table 6.1: Similarity values for target videos

	Mobile 352 × 288 (CIF)		Football 352 × 240 (SIF)		Table tennis 352 × 240 (SIF)		Foreman 352 × 288 (CIF)		Garden 352 × 240 (SIF)	
	Mark	No mark	Mark	No mark	Mark	No Mark	Mark	No Mark	Mark	No Mark
PSNR	35.2322	-	35.5779	-	35.7389	-	35.2744	-	35.9799	-
T	0.7323	0.0015	0.7142	0.0098	0.6769	-0.0084	0.8061	0.0025	0.6580	0.0189
Aspect to 4/3	0.7439	-0.0060	0.7477	-0.0181	0.7708	-0.0077	0.7621	-0.0029	0.7000	-0.0080
11/9	-	-	0.7555	-0.0168	0.7688	-0.0092	-	-	0.7014	-0.0084
16/9	0.6693	-0.0052	0.6872	0.0006	0.6902	-0.0082	0.6974	-0.0028	0.6213	-0.0081
Swap	0.6524	-0.0151	0.6866	0.0287	0.7224	-0.0454	0.7720	-0.0186	0.6154	-0.0149
Loss	0.7377	0.0120	0.7522	-0.0121	0.7660	-0.0072	0.7615	0.0137	0.7003	-0.0102
LP	0.6270	0.0134	0.5870	-0.0196	0.5685	0.0128	0.5725	-0.0067	0.5569	-0.0692
Light	0.7421	0.0008	0.7555	-0.0168	0.6335	0.0013	0.7612	-0.0029	0.6528	-0.0054
H	0.7444	-0.0059	0.7257	-0.0730	0.6960	-0.0407	0.7701	-0.0111	0.6998	-0.0221
R 0°	0.7421	0.0008	0.7540	-0.0168	0.7702	0.0013	0.7612	-0.0029	0.7054	-0.0054
1°	0.7331	-0.0001	0.7403	-0.0309	0.7597	0.0003	0.7536	-0.0033	0.6954	-0.006
2°	0.7320	0.0013	0.7400	-0.0221	0.7549	0.0011	0.7533	-0.0022	0.6938	-0.013
3°	0.7327	0.0008	0.7393	-0.0163	0.7538	0.0020	0.7489	-0.0036	0.6936	-0.0109
4°	0.7301	0.0020	0.7346	-0.0162	0.7516	0.0012	0.7448	-0.0028	0.6952	-0.0029
5°	0.7319	0.0033	0.7382	-0.0146	0.7478	0.0010	0.7439	-0.0028	0.6978	-0.0137
10°	0.7313	-0.0022	0.7060	0.0048	0.6948	0.0013	0.7320	-0.0033	0.7439	-0.0026
15°	0.7238	0.0025	0.7217	0.0116	0.6814	0.0046	0.7174	0.0007	0.7309	-0.0202
20°	0.7241	-0.0083	0.7108	0.0056	0.6914	-0.0002	0.7149	-0.0033	0.7303	-0.0079
25°	0.7135	-0.0053	0.7183	0.0139	0.6679	0.0012	0.7071	-0.0027	0.7240	-0.0102
30°	0.7034	0.0002	0.7236	0.0012	0.6605	0.0081	0.6863	-0.0047	0.6995	-0.0081
35°	0.7008	-0.0067	0.7231	0.0043	0.6753	0.0050	0.6836	-0.0014	0.6969	-0.0076
40°	0.6920	-0.0001	0.7234	0.0129	0.6690	-0.0001	0.6774	-0.0005	0.6894	-0.0096
45°	0.6846	-0.0112	0.7073	0.0026	0.6686	0.0052	0.6826	-0.0024	0.6963	-0.0006

6.6.2 Threshold

Since we use normalized correlation as the detection measure, two methods estimate the false positive probability [112]. The threshold T can be set according to proper false positive probability.

The first method is the approximate Gaussian method as follows:

$$P_{Gau} = \frac{1}{\sqrt{2\pi}\sigma_0} \int_T^{\infty} e^{\frac{-(x-m_0)^2}{2{\sigma_0}^2}} dx = Q(\frac{T - m_0}{\sigma_0}) \tag{6.26}$$

where, mean m_0 of the watermark sequence is zero because it is noise-like signal and uncorrelated to the original frame. $\sigma_0 = 1/\sqrt{n}$ is the standard deviation and n is the length of the watermark sequence. T is the threshold, and Q is the complementary

error function ($erfc$) and is defined as

$$Q(X) = \frac{1}{\sqrt{2\pi}} \int_X^\infty e^{\frac{-(x)^2}{2}} dx \tag{6.27}$$

This method models the distribution of normalized correlations as a Gaussian. With relatively low thresholds, this method is quite accurate. However, with higher thresholds, it shows overestimating the probability of false detections [112].

We choose the second method, developed by Miller et. al. [112]. The probability of a false detection is given by

$$P_{fpp} = \frac{I_{n-2}(T_\alpha)}{2I_{n-2}(\frac{\pi}{2})} \tag{6.28}$$

where,

$$T_\alpha = cos^{-1}T \tag{6.29}$$

$$I_d(\theta) = \int_0^\theta sin^d(u)du \tag{6.30}$$

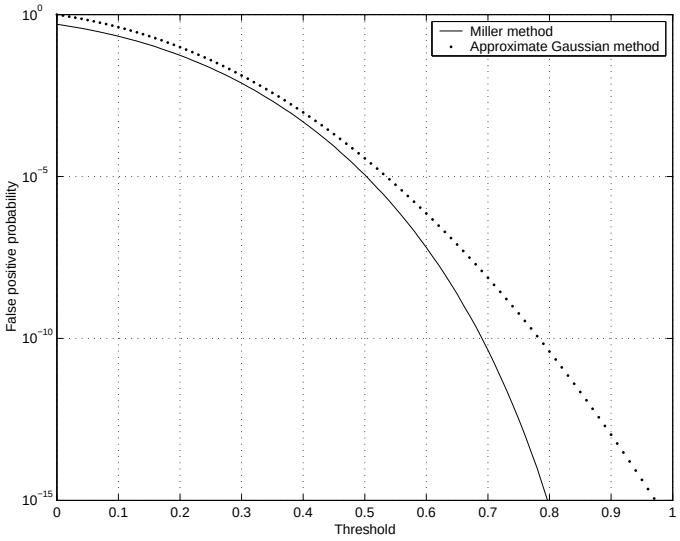


Figure 6.16: False positive probability estimation by Miller's method and approximate Gaussian method with the length of watermark sequence of 64.

False positive probability estimation by Miller's method and approximate Gaussian method is illustrated in Fig. 6.16. According to Miller's method, if the threshold is set to 0.4, 0.5 or 0.6, the false positive probability is 4.85×10^{-4}, 1.13×10^{-5}, or 6.45×10^{-8}. The higher the threshold is set, the smaller the probability of the false positive error will happen. We set our threshold as 0.5 according to our experimental results, for which the false positive probability of 1.13×10^{-5} is achieved.

6.6.3 Rotation with cropping

Normally, the rotation is very slight for video signals. The rotation angles are not more than 5°. Table 6.1 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 is much higher than the one for unwatermarked video. For very slight rotation up to 5°, the similarity values for watermarked video is much higher than the one for unwatermarked video. Even up to 45°, the similarity values for watermarked video and the values for unwatermarked video still can be successfully separated. Table 6.1 suggests that the proposed algorithm is very robust to rotation.

6.6.4 Translation

Translation is another attack which needs re-synchronization. The results for translation are shown in Table 6.1. No matter how many pixels the watermarked video is translated, the similarity results for watermarked video are the same after re-synchronization. The phase-only filtering method can correctly find out the translation parameters.

6.6.5 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, 4/3, 11/9, and 16/9. Different aspect ratios have been applied to the target video signals than their original ratio. The results are also shown in Table 6.1. In Table 6.1, "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.

6.6.6 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 6.1.

6.6.7 Frame loss

In this experiment, we drop one frame, and borrow one frame from the next GOP. As shown in row "Loss" in Table 6.1, we can clearly differentiate the watermarked video and the unwatermarked video.

6.6.8 Spatial filtering

As shown in row "LP" of Table 6.1, our algorithm is also robust to Gaussian noise addition with variance of 0.001 and Wiener low pass filtering process.

6.6.9 Light changing

The row "Light" in Table 6.1 shows that our algorithm is also robust to light change. In the table, the original frames are brightened by adding 50 to the gray scale values of the luminance component of each pixel.

6.6.10 Histogram equalization

Histogram equalization is used to spread out the most frequent intensity values and increase the image contrast. We apply histogram equalization to the target videos and the simulation results are shown our algorithm is robust to this attack in Table 6.1 in the row "H".

Table 6.2: Experimental results for MPEG-2 compression

	Mobile		Football		Table tennis		Foreman		Garden	
	Mark	No mark	Mark	No mark	Mark	No Mark	Mark	No Mark	Mark	No Mark
PSNR Y (dB)	18.62	19.35	21.31	21.66	22.16	21.04	22.13	23.01	17.66	18.09
U (dB)	24.70	25.49	27.68	28.59	33.80	33.20	32.88	32.92	23.41	24.74
V (dB)	23.89	24.87	32.52	33.65	28.07	28.01	31.28	32.55	27.10	28.94
Bitrate (kbps)	200	200	200	200	200	200	200	200	335	335
MPEG-2	0.5622	0.2046	0.6048	0.0809	0.5249	0.1480	0.5448	0.2642	0.5210	0.2087

Table 6.3: Experimental results for H.264 compression

	Mobile		Football		Table tennis		Foreman		Garden	
	Mark	No mark	Mark	No mark	Mark	No Mark	Mark	No Mark	Mark	No Mark
PSNR Y (dB)	26.33	27.54	26.29	27.94	27.68	28.90	34.29	35.09	27.36	27.90
U (dB)	33.60	34.79	33.52	34.23	37.67	38.62	39.82	39.92	33.58	33.70
V (dB)	33.40	34.77	35.79	36.95	36.32	37.25	42.65	42.56	33.58	33.89
Bitrate (kbps)	569	665	372	480	180	230	331	399	586	702
H.264	0.5554	0.1314	0.5589	-0.0695	0.5463	0.1878	0.5167	0.0327	0.5102	0.1067

6.6.11 MPEG-2 compression

Table 6.2 shows the simulation results for MPEG-2 compression. Our algorithm could be robust to MPEG-2 with the bit rate as low as 200 kbps which represents bad quality. PSNRs in this table show the quality after MPEG-2 compression. Lower PSNR means worse quality because more redundant information is removed by the compression algorithm.

6.6.12 H.264 compression

Table 6.3 shows the simulation results for H.264 compression. This table first illustrates that H.264 provides more efficient compression than MPEG-2. For example, for video "$Table - tennis$", the quality (PSNR of 27.68 dB) after H.264 compression with lower bit rates (180 kbps) is better than the quality (PSNR of 22.61 dB) after MPEG-2 compression with higher bit rates (200 kbps). Second, this table shows that our algorithm is robust to H.264 compression with low bit rates. Third, for the same video, in order to get the similar detection results for the watermarked video after MPEG-2 compression and H.264 compression, we have to use H.264 encoded watermarked video with the better quality (higher PSNR) than MPEG-2 encoded video. It approves that H.264 is the more efficient compression standard and removes more redundant information than MPEG-2.

6.6.13 Combinational 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

Table 6.4: Experimental results for combinational attacks (RST and MPEG-2)

	Mobile		Football		Table tennis		Foreman		Garden	
	Mark	No mark	Mark	No mark	Mark	No Mark	Mark	No Mark	Mark	No Mark
PSNR Y (dB)	20.85	20.91	23.40	24.30	25.32	26.51	28.10	29.01	21.68	22.54
U (dB)	27.20	26.23	29.11	29.65	33.84	34.85	34.24	35.22	25.63	26.31
V (dB)	26.73	26.09	32.76	33.76	30.59	31.28	34.52	35.65	29.06	30.43
Bitrate (kbps)	1200	1200	975	975	1000	1000	1400	1400	2300	2300
R 0°	0.5724	0.0858	0.6036	0.0128	0.6109	0.0995	0.5760	0.0340	0.5928	0.0052
1°	0.5416	0.0727	0.5696	-0.0238	0.6019	0.0970	0.5506	0.0329	0.5604	0.0079
2°	0.5300	0.0740	0.5712	-0.0146	0.5389	0.0807	0.5471	0.0313	0.5421	0.0093
3°	0.5237	0.0788	0.5403	-0.0128	0.5547	0.0862	0.5326	0.0323	0.5369	0.0090
4°	0.5233	0.0876	0.5347	-0.0076	0.5464	0.0818	0.5301	0.0341	0.5405	0.0051
5°	0.5173	0.0729	0.5379	-0.0088	0.5589	0.0855	0.5279	0.0325	0.5394	0.0119
6°	0.5210	0.0705	0.5336	-0.0046	0.5455	0.0798	0.5256	0.0088	0.5373	0.0084
7°	0.5114	0.0726	0.5364	-0.0045	0.5432	0.0877	0.5126	0.0067	0.5366	0.0215
8°	0.5102	0.0803	0.5367	-0.0028	0.5338	0.0913	0.5121	0.0092	0.5314	0.0140
9°	0.5013	0.0549	0.5402	0.0013	0.5377	0.0720	0.5062	0.0110	0.5244	0.0154
10°	0.5022	0.0701	0.5275	-0.0055	0.5407	0.0740	0.5035	0.0094	0.5335	0.0182

following combined attacks – MPEG2 or H.264 compression, Gaussian noise, rotation, aspect ratio change, translation, lighting change – to the target videos. The simulation results are shown in Table 6.4 and Table 6.5. In Table 6.4 and Table 6.5, "R" means rotation.

The bit rates for MPEG-2 compression or H.264 compression we used here have to be higher than the ones in Table 6.2 and Table 6.3 for compression alone. The variance of the additive Gaussian noise is 0.001. Rotation angles are from 0° to 10°. The aspect ratio changes are from the original to 16/9 for each target video. Translations are 20 pixels to the right. The light changing is to increase the luminance by 50.

Table 6.4 shows the simulation results for the combinational attacks including RST, MPEG-2 compression, noise and lighting change. With normalized cross-correlation as our detection method, changing the brightness of the target frames has no affect on the simulation results. Therefore, increasing or decreasing the gray scale has the same results for our algorithm. Table 6.5 shows the simulation results for the combination

Table 6.5: Experimental results for combinational attacks (RST and H.264)

	Mobile		Football		Table tennis		Foreman		Garden	
	Mark	No mark	Mark	No mark	Mark	No Mark	Mark	No Mark	Mark	No Mark
PSNR Y (dB)	27.20	28.09	28.65	30.11	33.12	34.06	37.96	38.65	33.22	34.76
U (dB)	35.14	36.44	33.50	35.20	39.77	39.92	42.04	43.43	35.14	36.52
V (dB)	34.76	35.92	35.15	36.95	39.09	40.06	45.28	46.74	35.66	36.92
Bitrate (kbps)	1114	1231	908	1125	917	1084	835	965	1839	1903
R 0°	0.6449	-0.0016	0.6288	-0.0052	0.6135	0.0115	0.6230	0.0096	0.5950	-0.0072
1°	0.6103	-0.0044	0.5747	-0.0033	0.5767	0.0125	0.5893	0.0088	0.5525	-0.0105
2°	0.5989	-0.0045	0.5905	-0.0412	0.5530	0.0031	0.5909	0.0067	0.5428	-0.0037
3°	0.5991	-0.0100	0.5692	-0.0311	0.5538	0.0059	0.5472	0.0092	0.5480	0.0001
4°	0.5856	-0.0027	0.5613	-0.0302	0.5383	0.0079	0.5461	0.0110	0.5346	-0.0064
5°	0.5859	-0.0014	0.5676	-0.0302	0.5391	0.0057	0.5495	0.0094	0.5356	0.0040
6°	0.5830	-0.0136	0.5641	-0.0293	0.5339	0.0104	0.5401	0.0088	0.5102	0.0061
7°	0.5880	-0.0076	0.5693	-0.0336	0.5334	0.0108	0.5345	0.0098	0.5000	0.0077
8°	0.5861	-0.0177	0.5669	-0.0277	0.5194	0.0081	0.5353	0.0110	0.5080	-0.0061
9°	0.5826	-0.0054	0.5724	-0.0273	0.5230	0.0105	0.5330	0.0083	0.5235	-0.0012
10°	0.5891	-0.0046	0.5604	-0.0198	0.5254	0.0153	0.5275	0.0093	0.5249	0.0088

attacks including RST, H.264 compression, noise and lighting change. Both Table 6.4 and Table 6.5 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 combinational attacks as mentioned above.

6.6.14 Performance comparison

In this section, we provide some discussions on our experimental results by comparing to the two existing video watermarking algorithms [58][62] 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 correlation value for Lu's algorithm [58] 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 [62]. 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.

In Section 3.2.1, we review the algorithms robust to MPEG-2 compression. By using error control coding (ECC), such as, BCH code or interleaving, researchers could increase the robustness of their algorithms. One of the most advanced algorithm robust to MPEG-2 compression was proposed by Shao et al. [52]. They could successfully decode the watermark when MPEG-2 compression is down to 200 kbps without any other attacks and 300 kbps + scaling to 50%. ECC can only work with binary watermark pattern, and cannot work with noise like watermark pattern, such as ours. Even though, our algorithm also can be robust to MPEG-2 compression with bit rate of 200 kbps.

The video algorithms we reviewed in Section 3.2.3 all work within H.264 encoding process. None of current video algorithms claim to be robust to both geometrical attacks and H.264 compression. Our algorithm could robust to H.264 plus combinational attacks with average PSNR of 32 dB after compression, which is a reasonable quality for SIF or CIF video.

Chapter 7

Conclusions and future work

This thesis focused on video watermarking algorithms robust against geometric attacks and video compressions. After introducing the general information and techniques of video watermarking, we reviewed existing algorithms robust against geometric attacks and analyzed the methods or techniques used by the recent video watermarking algorithms to provide robustness against video compressions like MPEG-2, MPEG-4 and H.264. The conclusions we drew from our review are, 1). geometric attacks are the big challenge to video watermarking algorithms due to synchronization loss, and 2). H.264 is the most efficient video compression standard that can defeat video watermarking algorithms due to its effective compression techniques. To the best of our knowledge, no existing video watermarking algorithm claims to be robust to both RST attacks and H.264 compression. Our motivation is to develop a blind video watermarking algorithm to provide robustness against both RST attacks and H.264 compression.

We first extended our RST invariant image watermarking algorithm to I-frame of each GOP of video signals to develop a robust video watermarking algorithms (LPM-POF). This method is based on the log-polar mapping and our new phase-only filtering

method. The key theory for this algorithm is that rotation and scaling of an image in spatial domain converts to a horizontal and vertical translation in the log-polar domain of Fourier magnitude of the target image. We use a matching template to find the embedding location of watermark pattern for watermark detection. This method is very robust to RST attacks, common signal processing and MPEG-2. However, it does not work well for H.264 compression.

The LPMPOF algorithm can be used as a RST parameters detector to enhance the robustness against geometric attacks for other video watermarking algorithms. The DCTLSB algorithm is a semi-fragile video watermarking algorithm, which has a high capacity and low robustness against geometric attacks. We integrate our LPMPOF algorithm with it to improve its robustness against RST attacks and keep the high capacity as well. A convolutional coding is used to improve the error bit rate.

Then, we proposed a novel video watermarking algorithm based on the 1D DFT and Radon transform. The 1D DFT of a GOP provides an ideal domain, in which both the spatial information and the temporal frequency information can be obtained. We embed the watermark in the optimal selected area to avoid the locations with high spatial frequency and low temporal frequency because these parts will be significantly removed by compression processing. Finally, we choose frames with the highest temporal frequency to embed the adaptive fence-shaped watermark pattern, which could re-synchronize frames having undergone RST attacks and keep the fidelity of the watermarked video. The I-frame of each GOP always has more watermark information than other frames and this could improve the robustness 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.

We mainly proposed two video watermark algorithms in this thesis. Each of them has its own advantage and drawback. The first one, which is log-polar based, is very robust against RST attacks. However, the robustness against H.264 compression is weak. The second algorithm, which is 1D DFT based, is very robust against RST attacks (although weaker than the first one) and H.264.

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.

Bibliography

[1] C. Lin, M. Wu, J. Bloom, I. Cox, M. Miller, and Y. Lui, "Rotation, scale, and translation resilient watermarking for images," *IEEE Transactions on Image Processing*, vol. 10, no. 5, pp. 767–782, 2001.

[2] Y. Wang, J. Ostermann, and Y. Zhang, *Waveform-Based Video Coding*. Signal Processing, Prentice Hall, 2001.

[3] I. Cox, M. Miller, and J. Bloom, *Digital Watermarking*. ISBN: 1-55860-714-5, Morgan Kaufmann Publishers, 2002.

[4] G. Doerr and J.-L. Dugelay, "A guide tour of video watermarking," *IEEE Signal Processing: Image Communication*, vol. 18, no. 4, pp. 263–282, 2003.

[5] J. O'Ruanaidh and T. Pun, "Rotation, scale, and translation invariant digital image watermarking," *Signal Processing*, vol. 66, no. 3, pp. 303–317, 1998.

[6] S. Winkler, *Digital Video Quality: Vision Models and Metrics*. Wiley, 2005.

[7] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[8] G. C. Langelaar and R. L. Lagendijk, "Optimal differential energy watermarking of DCT encoded images and video," *IEEE Transactions on Image Processing*, vol. 10, no. 1, pp. 148–158, 2001.

[9] F. Deguillaume, G. Csurka, J. J. K. O'Ruanaidh, and T. Pun, "Robust 3D DFT video watermarking," in *Proceedings of IS&T/SPIE Electronic Imaging 99*, vol. 3657, pp. 113–124, 1999.

[10] P. Chan and 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*, pp. 202–213, 2003.

[11] G. Depovere, T. Kalker, J. Haritsma, M. Maes, L. D. Strycker, P. Termont, J.Vandewege, A. Langell, C. Alm, P. Normann, G.O'Reilly, B.Howes, H. Vaanholt, R.Hintzen, P. Donnely, and A. Hudson, "The VIVA project: digital watermarking for broadcarst monitoring," in *Proceedings of IEEE International Conference on Image Processing*, vol. 2, pp. 202–205, 1999.

[12] T. Kalker, G. Depovere, J. Haitsma, and M. Maes, "A video watermarking system for broadcast monitoring," in *Proceedings of SPIE, Security and Watermarking of Multimedia Content*, vol. 3657, pp. 103–112, 1999.

[13] J. Linnartz, "The ticket concept for copy control based on embedded signalling," in *Proc. of the fifth European Symposium on Research in Computer Security, Lecture Notes in Computer Science*, vol. 1485, pp. 257–274, 1998.

[14] B. Mobasseri, M. Sieffert, and R. Simard, "Contect authentication and tamper detection in digital video," in *Proceedings of IEEE International Conference on Image Processing*, vol. 1, pp. 458–461, 2000.

[15] B. Mobasseri, "Direct sequence watermarking of digital video using m-frames," in *Proceedings of IEEE International Conference on Image Processing*, vol. 3, pp. 399–403, 1998.

[16] J. Dittmann, A. Steinmetz, and R. Steinmetz, "Content-based digital signature for motion pictures authentication and content fragile watermarking," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, vol. 2, pp. 209–213, 1999.

[17] M. Veen, A. Lemma, M. Celik, and S. Katzenbeisser, *Forensic Watermarking in Digital Rights Management*. Data-Centric Systems and Applications, Springer Berlin Heidelberg, 2007.

[18] H. Cheng and M. Isnardi, "Spatial, temporal and histogram video registration for digital watermark detection," in *Proceedings of IEEE International Conference on Image Processing*, vol. 2, pp. II–735–II–738, 2003.

[19] P. Nguyen, R. Balter, N. Montfor, and S. Baudry, "Registration methods for non blind watermark detection in digital cinema applications," in *Proceedings of SPIE/IST Electronic Imaging*, vol. 5020, pp. 553–562, 2003.

[20] D. Delannay, C. de Roover, and B. Macq, "Temporal alignment of video sequences for watermarking systems," in *Proceedings of SPIE/IST Electronic Imaging*, vol. 5020, pp. 481–492, 2003.

[21] F. Bartolini, A. manetti, A. Piva, and M. Barni, "A data hiding approach for correcting errors in H. 263 video transmitted over a noisy channel," in *Proceedings of the IEEE Fourth Workshop on Mulitmedia Signal Processing*, pp. 65–70, 2001.

[22] D. Mukherjee, J. Chae, and S. Mitra, "A source and channel coding approach to data hiding with applications to hiding speech in video," in *Proceedings of IEEE International Conference on Image processing*, vol. 1, pp. 348–352, 1998.

[23] M. Swanson, B. Zhu, and A. Tewfik, "Data hiding for video-in-video," in *Proceedings of IEEE International Conference on Image processing*, vol. 2, pp. 676–679, 1997.

[24] Y. Wang, J. Ostermann, and Y. Zhang, *Video Compression Standards*. Signal Processing, Prentice Hall, 2001.

[25] D. Hearn and M. Baker, *Computer graphics*. ISBN:0-13-530924-7, Prentice Hall, 1997.

[26] Y. Wang, J. Ostermann, and Y. Zhang, *Video Formation, Perception, and Represenatation*. Signal Processing, Prentice Hall, 2001.

[27] R. Gonzalez and R. Woods, *Digital Image Processing*. ISBN: 0-20-118075-8, Prentice Hall, 2002.

[28] J. Horner and P. Gianino, "Phase-only matched filter," *Applied Optics*, vol. 23, no. 6, pp. 812–816, 1984.

[29] D. Zheng, J. Zhao, and 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*, vol. 13, no. 8, pp. 753–765, 2003.

[30] R. Bracewell, *The Fourier Transform and Its Applications*. ISBN: 0073039381, Boston : McGraw Hill, 2000.

[31] H. Kim, Y. Baek, H. Lee, and Y. Suh, "Robust image watermark using Radon transform and bispectrum invariants," *LNCS 2578*, pp. 145–159, 2003.

[32] D. Simitopoulos, D. E. Koutsonanos, and M. G. Strintzis, "Robust image watermarking based on generalized radon transformations," *IEEE Transcations on Circuits and System for Video technology*, vol. 13, no. 8, pp. 732–745, 2003.

[33] Y. Hsu and H. Arsenault, "Optical pattern recognition using circular harmonic expansion," *Applied Optics*, vol. 21, pp. 4012–4015, 1982.

[34] J. Yao and G. Letreton, "Scale-invariant correlation with truncated phase-only radial harmonic fitlers," *Optics Communication 145*, pp. 213–219, 1998.

[35] J. Yao and L. Chin, "Power-adjusted fractional power radial harmonic filters for shift- and scale-invariant pattern recognition with improved noise robustness and discrimination," *Optics Communication 1162*, pp. 26–30, 1998.

[36] H. Kim and B. Kumar, "Rotation-tolerant watermark detection using circular harmonic function correlation filter," *Digital Watermarking, LNCS 2939*, pp. 263–276, 2004.

[37] M. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. IT-8, no. 8, pp. 1409–1420, 1962.

[38] C. Harris and M. Stephen, "A combined corner and edge detector," in *Proceeding of 4th Alvey Vision Conference*, pp. 147–151, 1988.

[39] D. Simitopoulos, D. Koutsonanos, and M. G. Strintzis, "Image watermarking resistant to geometric attacks using generalized radon transformations," in *Proceedings of DSP 2002*, vol. 1, pp. 85–88, 2002.

[40] T. Wiegand, G. J.Sullian, G. Bjontegaard, and A.Luthra, "Overview of the H.264 video coding standard," *IEEE Transaction on Circuits System and Video Technology*, vol. 13, pp. 560–576, 2003.

[41] M. Kutter, "Watermarking resistant to translation, rotation, and scaling," in *Proceedings of SPIE*, vol. 3628, pp. 423–431, 1998.

[42] A. Herrigel, J. J. K. O'Ruanaidh, H. Petersen, S. Pereira, and T. Pun, "Secure copyright protection techniques for digital images," in *Proceedings of Int. Workshop on Information Hiding*, 1998.

[43] S. Pereira, J. O'Ruanaidh, and F. Deguillaume, "Template based recovery of fourier-based watermarks using log-polar and log-log maps," in *Proceedings of IEEE International Conference on Multimedia Computing and System*, vol. 1, pp. 870–874, 1999.

[44] G. Csurka, F. Deguillaume, J. J. K. O'Ruanaidh, and T. Pun, "A Bayesian approach to affine transformation resistant image and video watermarking," in *Proceedings of Int. Workshop on Information Hiding*, vol. 2, pp. 315–330, 1999.

[45] S. Pereira and T. Pun, "Robust template matching for affine resistant image watermarks," *IEEE Transactions on Image Processing*, vol. 9, no. 6, pp. 1123–1129, 2000.

[46] R. Caldelli, M. Barni, and A. Piva, "Geometric-invariant robust watermarking through constellation matching in the frequency domain," in *Proceedings of IEEE International Conference on Image Processing*, vol. 2, pp. 65–68, 2000.

[47] D. Zheng, Y. Liu, J. Zhao, and A. El Saddik, "A survey of RST invariant image

watermarking algorithms," *ACM Computing Surveys*, vol. 39, no. 2, pp. 1–91, 2007.

[48] F. D. S. Voloshynovskiy and T. Pun, "Multibit digital watermarking robust against local nonlinear geometrical distortions," in *IEEE International Conference on Image Processing, ICIP 2001*, pp. 999–1002, 2001.

[49] S.Voloshynovskiy, F.Deguillaume, and T.Pun, "Content adaptive watermarking based on a stochastic multiresolution image modeling," in *EUSIPCO2000, X Europian Signal Processing Conference*, September 4-8, 2000.

[50] Y. Zhao and R. L. Lagendijk, "Video watermarking scheme resistant to geometric attacks," in *Proceedings of IEEE Conference on Image Processing*, vol. 2, pp. 145–148, 2002.

[51] R. Lancini, F. Mapelli, and S. Tubaro, "A robust video watermarking technique in the spatial domain," in *IEEE Region-8 International Symposium on Video/Image Processing and Multimedia Communications*, pp. 251–256, 2002.

[52] Y. Shao, L. Zhang, G. Wu, and X. Lin, "A novel frequency domain watermarking algorithm with resistance to geometric distortions and copy attack," in *Proceedings of IEEE International Symposium on Circuits Systems*, vol. 2, pp. 940–943, 2003.

[53] Y. Liu, D. Wang, and J. Zhao, "Video watermarking based on scene detection and 3D DFT," in *IASTED International Conference on Circuits, Signals and Systems*.

[54] J. Haitsma and T. Kalker, "A watermarking scheme for digital cinema," in *Pro-*

ceedings of IEEE International Conference on Image Processing, vol. 2, pp. 487–489, 2001.

[55] M. Dainaka, S. Nakayama, I. Echizen, and H. Yoshiura, "Dual-plane watermarking for color pictures immune to rotation, scale, translation, and random bending," in *Proceedings of IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 93–96, 2006.

[56] I. Echizen, Y. Atomori, S. Nakayama, and H. Yoshiura, "Use of human visual system to improve video watermarking for immunity to rotation, scale, translation, and random distortion," *Circuits, Systems, and Signal Processing*, vol. 27, no. 2, pp. 213–227, 2008.

[57] R. Kumar, H. S. Sawhney, J. C. Asmuth, A. Pope, and S. Hsu, "Registration of video to geo-referenced imagery," in *Proceedings of 14th International Conference on Pattern Recognition (ICPR'98)*, vol. 2, p. 1393, 1998.

[58] C.-S. Lu and 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, pp. 483–486, 2001.

[59] C. T. Hsu and Y. S. Tsai, "Video object watermarking by quadratic region decomposition and tessellation," *Images and Recognition*, vol. 9, no. 1, March 2003.

[60] N. V. Boulgouris, F. D. Koravos, and M. G. Strintzis, "Self-synchronizing watermark detection for MPEG-4 objects," in *Proceedings of IEEE Conference on Electronics, Circuits, and Systems*, vol. 3, pp. 1371–1374, 2001.

[61] P. Bas, J. Chassery, and B. Marq, "Geometrically invariant watermarking using

feature points," *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 1014–1028, 2002.

[62] Y. Wang and A. Pearmain, "Blind MPEG-2 video watermarking robust against geometric attacks: A set of approaches in DCT domain," *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1536–1543, 2006.

[63] M. El'arbi, C. B. Amar, and H. Nicolas, "A video watermarking scheme resistant to geometric transformations," in *Proceedings of IEEE International Conference on Image Processing*, vol. 5, pp. 481–484, 2007.

[64] H. Zhuang, Y. Li, and C. Wu, "A blind spatial-temporal algorithm based on 3D wavelet for video watermarking," in *Proceedings of IEEE International Conference on Multimedia and Expo*, vol. 3, pp. 1727–1730, 2004.

[65] K. Su, D. Kundur, and D. Hatzinakos, "A content dependent spatially localized video watermark for resistance to collusion and interpolation attacks," in *Proceedings of IEEE International Conference on Image Processing*, vol. 1, pp. 818–821, 2000.

[66] H. Liu, N. Chen, J. Huang, X. Huang, and Y. Q. Shi, "A robust DWT-based video watermarking algorithm," in *Proceedings of IEEE International Symposium on Circuits Systems*, vol. 3, pp. 631–634, 2002.

[67] B. Kim, J. Choi, and K. Park, "RST-resistant image watermarking using invariant centroid and reordered Fourier-Mellin transform," in *Proceedings of IWDW' 2003, LNCS 2939*, pp. 370–381, 2004.

[68] Y. Liu, D. Zheng, and J. Zhao, "A rectification scheme for RST invariant image

watermarking," *IEICE Transactions on Fundamentals, Special Section on Cryptography and Information Security*, vol. E88-A, no. 1, pp. 314–318, 2005.

[69] C. H. Teh and R. T. Chin, "On image analysis by the method of moments," *IEEE Trans. PAMI*, vol. 10, no. 4, pp. 496–513, 1988.

[70] Y. Xin, S. Liao, and M. Pawlak, "Geometrically robust image watermarking via pseudo zernike momens," in *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2004.

[71] M. Alghoniemy and A. H. Tewfik, "Geometric invariance in image watermarking," *IEEE Transactions on Image Processing*, vol. 13, no. 2, pp. 145–153, 2004.

[72] Y. Abu-Mostafa and D. Psaltis, "Recognitive aspects of moment invariants," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 698–706, 1984.

[73] A. Khotanzad and Y. Hong, "Invariant image recognition by zernike moments," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 489–497, 1990.

[74] M. Farzam and S. Shirani, "A robust multimedia watermarking technique using Zernike transform," in *Proc. of IEEE International Workshop Multimedia Signal Processing*, pp. 529–534, 2001.

[75] M. Pawlak and Y. Xin, "Robust image watermarking: an invariant domain approach," in *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, vol. 2, pp. 885–888, 2002.

[76] P. Dong and N. P. Galatsanos, "Affine transformation resistant watermarking based on image normalization," in *Proceedings of IEEE International Conference on Image Processing*, vol. 3, pp. 489–492, 2002.

[77] H. S. Kim and H.-K. Lee, "Invariant image watermark using zernike moments," *IEEE Transactions on Circuits and System for Video technology*, vol. 13, no. 8, pp. 766–775, 2003.

[78] V. Chandran, B. Carswell, B.Boashash, and S. Elgar, "Pattern recognition using invariants defined from higher order spectra: 2-D image inputs," *IEEE Trans. on Image Processing*, vol. 6, no. 5, pp. 703–712, 1997.

[79] H. Kim, Y. Baek, and H. Lee, "Rotation, scale and translation invariant image watermark using higher order spectra," *Optica Engineering*, vol. 42, no. 2, pp. 340–349, 2003.

[80] X. Dai and S. Khorram, "A feature-based image registration algorithm using improved chain-code representation combined with invariant moments," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 5, pp. 2351–2362, 1999.

[81] M. Alghoniemy and A. H. Tewfik, "Image watermarking by moment invariants," in *Proceedings of IEEE International Conference on Image Processing*, vol. 2, pp. 73–76, 2000.

[82] B. Abdel-Aziz, J. Zhao, and J.-Y. Chouinard, "On the limits of second generation watermarks," in *Proceeding of IEEE Canadian Conference on Computer and Robot Vision (CRV2004)*, pp. 209–216, 2004.

[83] Y. Liu and J. Zhao, "A robust image watermarking method based on adaptive feature points detection," in *Proc. of IEEE International Conference on Communications, Circuits and Systems*.

[84] C. W. Tang and H. M. Hang, "A feature-based robust digital image watermarking scheme," *IEEE Transactions on Signal Processing*, vol. 51, no. 4, APRIL 2003.

[85] M. Kutter, S. Bhattacharjee, and T. Ebrahimi, "Towards second generation watermarking schemes," in *Proc of IEEE International Conference on Image Processing '99*, vol. I, pp. 320–323, 1999.

[86] Z. Duric, N. Johnson, and S. Jajodia, "Recovering watermarks from images," in *Informaion & Software Engineering Technical Report ISE-TR-99-04*, 1999.

[87] Q. Sun, J. Wu, and R. Deng, "Recovering modified watermarked image with reference to original image," in *Proceedings of SPIE*, pp. 415–424, 1999.

[88] M. Alghoniemy and A. Tewfik, "Geometric distortion correction in image watermarking," in *Proceedings of SPIE*, pp. 82–89, 2000.

[89] J. Dittmann, T. Fiebig, and R. Steinmetz, "New approach for transformation-invariant image and video watermarking in the spatial domain: self-spanning patterns (ssp)," in *Proceedings of SPIE*, pp. 176–186, 2000.

[90] T.-T. Lu, W.-L. Hsu, and P.-C. Chang, "Blind video watermarking for H.264," in *Canadian Conference on Electrical and Computer Engineering*, pp. 2353–2356, 2006.

[91] G. Qiu, P. Marzilian, A. T. S. Ho, D. He, and Q. Sun, "A hybrid watermarking scheme for H.264 video," in *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 4, pp. 865–868, 2004.

[92] M. Noorkami and R. Mersereau, "Compressed domain video watermarking for

H.264," in *IEEE International Conference on Image Processing*, vol. 2, pp. 890–893, 2005.

[93] J. Zhang, A. T. S. Ho, G. Qiu, and P. Marziliano, "Robust video watermarking of H.264/AVC," *IEEE Transaction on Circuits and Systems*, vol. 54, no. 2, pp. 205–209, 2007.

[94] J.-Y. Hsieh and L.-W. Chang, "A high capacity watermarking for H.264/AVC based on frequency weighting," in *International Symposium on Intelligent Signal Processing and Communication Systems*, 2006.

[95] S. Sakazawa, Y. Takishima, and Y. Nakajima, "H.264 native video watermarking method," in *IEEE International Symposium on Circuits and Systems*, pp. 1439–1442, 2006.

[96] N. Mohaghegh and O. Fatemi, "H.264 copyright protection with motion vector watermarking," in *International Conference on Audio, Language and Image Processing*, pp. 1384–1389, 2008.

[97] M. A. Ali and E. A. Edirisinghe, "Watermarking H.264/AVC by modifying DC coefficients," in *2009 International Conference on CyberWorlds*, pp. 241–245, 2000.

[98] S. Lin and D. Costello, *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, Prentice Hall, 2004.

[99] J. Zhang and A. Ho, "An efficient digital image-in-image watermarking algorithm using the integer discrete cosine transform (IntDCT)," in *IEEE Joint Conferene of 4th International Conference on Information, Communication and Signal Processing and 4th Pacific-Rim Conference on Multimedia*, 2003.

[100] Y. Liu, D. Zheng, and J. Zhao, "An image rectification scheme and its applications in RST invariant digital image watermarking," *Springer Journal: Multimedia Tools and Applications*, vol. 34, no. 1, pp. 57–84, 2007.

[101] D. L. Flannery and J. L. Horner, "Fourier optical signal processors," *Processings of the IEEE*, vol. 77, no. 10, 1989.

[102] B. Reddy and B. Chatterji, "An FFT-based technique for translation, rotation and scale-invariant image registration," *IEEE Transactions on Image processing*, vol. 5, no. 8, 1996.

[103] L. Hill and T. Vlachos, "On the estimation of global motion using phase correlation for broadcast applications," *IEE Proceedings of Image Processing and Its Application*, vol. 465, no. 2, pp. 721–725, 1999.

[104] D. Zheng and J. Zhao, "RST invariant digital image watermarking: importance of phase information," in *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 785–788, 2003.

[105] C. Kuglin and D. Hines, "The phase correlation image alignment method," in *Proceedings of tThe IEEE 1975 International conference on cybernetrics and society (Sept.)*, pp. 163–165, 1975.

[106] J. Linnartz, T. Kaler, G. Depovere, and R. Beuker, "A reliability model for the detection of electronic watermarks in digital images," in *Proceedings of IEEE Fifth Symposium on Communication and Vehicular Technology*, pp. 202–209, 1997.

[107] G. Depovere, T. Kalker, and J. Linnartz, "Improved watermark detection using

filtering before correlation," in *Proceedings of IEEE International Conference on Image Processing*, pp. 430–434, 1998.

[108] C. yung Lin and S. fu Chang, "Issues and solutions for authenticating mpeg video," in *IEEE International Conference on Acoustics, Speach and Signal Processing*.

[109] S. B. Wicker, *Error control systems for digital communication and storage*. Englewood Cliffs, N.J. : Prentice Hall, 1995.

[110] Y. Liu and 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)*, pp. 443–448, 2008.

[111] A. M. Eskicioglu and P. S. Fisher, "Image quality measures and their performance," *IEEE Transactions on Communication*, vol. 43, p. 2959C2965, 1995.

[112] M. Miller and J. Bloom, "Computing the probability of false watermark detection," in *Proceedings of the Third International Workshop on Information Hiding*.