Motion-Coherent Video Watermarking

VINOD P

Motion-Coherent Video Watermarking

A

Thesis Submitted

in Partial Fulfilment of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

By

VINOD P

Department of Electronics and Communication Engineering

Indian Institute of Technology Guwahati

Guwahati - 781 039, INDIA.

July, 2007

Motion-Coherent Video Watermarking

A

Thesis Submitted

in Partial Fulfilment of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

By

VINOD P

Department of Electronics and Communication Engineering

Indian Institute of Technology Guwahati

Guwahati - 781 039, INDIA.

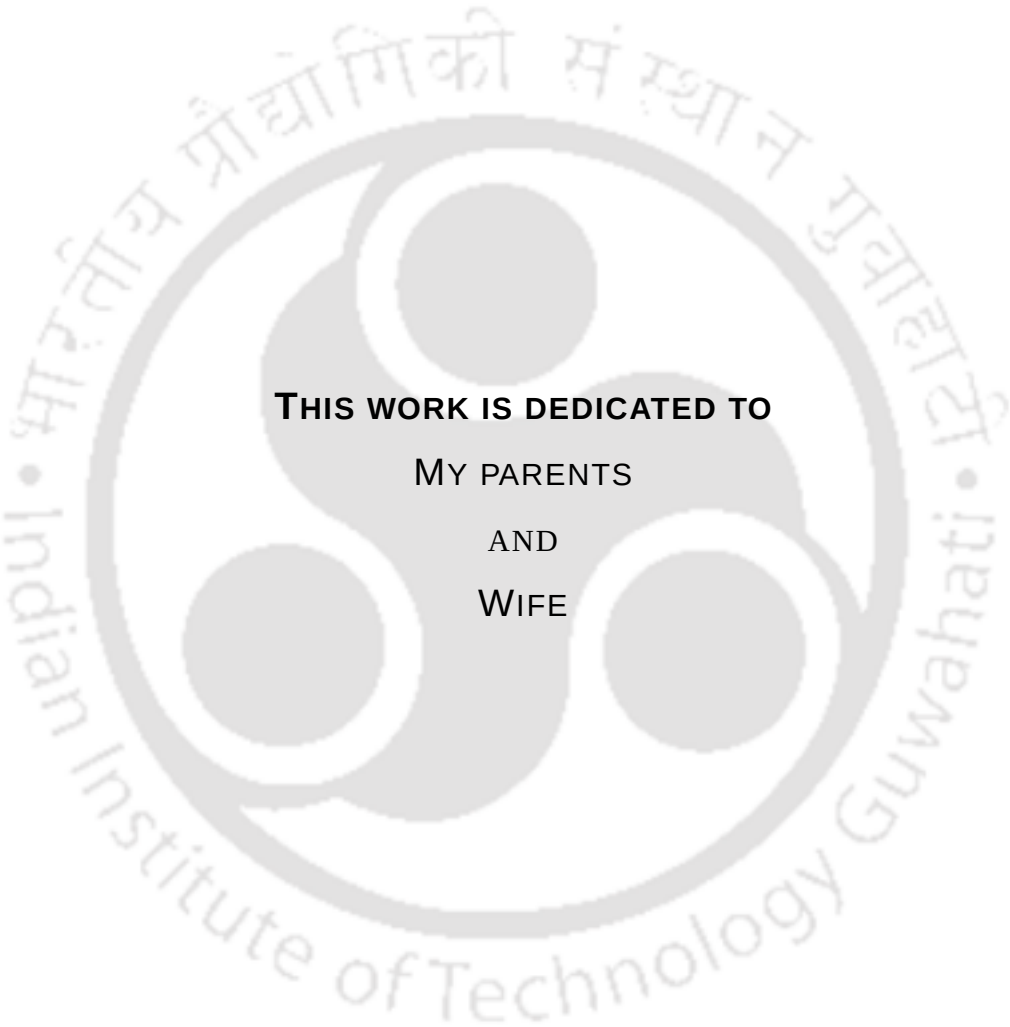
July, 2007

Certificate

This is to certify that the thesis entitled "**Motion-Coherent Video Watermarking**", submitted by Vinod P, a research scholar in the *Department of Electronics and Communication Engineering , Indian Institute of Technology Guwahati*, for the award of the degree of **Doctor of Philosophy**, is a record of an original research work carried out by him under my supervision and guidance. The thesis has fulfilled all requirements as per the regulations of the Institute and in my opinion has reached the standard needed for submission. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dated:
Guwahati.

Prof. P. K. Bora
Dept. of Electronics and Communication Engg.
Indian Institute of Technology Guwahati
Assam - 781 039
India.



THIS WORK IS DEDICATED TO

MY PARENTS

AND

WIFE

Acknowledgements

First of all, I would like to thank my supervisor Prof. P. K. Bora, for his valuable guidance, moral support and endless encouragement. I would also like to thank my doctoral committee members, Dr. S. Dandapat, Dr. J. S. Sahambi, Prof. A. Mahanta and Dr. D. Ghosh. Their suggestions and assistance have been valuable. I specially thank Dr. Gwenael Doerr (University College London), for the fruitful discussions and valuable suggestions. I am thankful to Mr. Sanjib Das and Mr. L. N. Sarma, research engineers in the department, for their help during the entire course of this work. I am grateful to all the other faculty members and staff of the department of ECE.

Among my friends, I would like to extend my special thanks to Ali, Babusena, Bitta, B. Deka, D. K. Gogoi, Manas, Manglem, Mrinal, V. K. Nath, Padhi, Ram Babu, N. Saikia, Senthil and Sivasankar. I would also like to thank Bajal, Natasha, Ram and Sai of the YFT.

My parents, wife, brother, and sister were always behind me, regardless of being miles away. Their love and confidence gave me strength when I needed it most. I would like to express my appreciation to my uncle, Mr. N. Vasu for supporting my decision to join PhD.

(Vinod P)

Abstract

One of the challenging security issues in video watermarking is the *inter-frame collusion attacks*. These attacks exploit the inherent redundancy in the video frames or in the watermark to produce an un-watermarked copy of the video. The *frame temporal filtering* (FTF) is the basic inter-frame collusion attack in which temporal low-pass filtering is applied to the watermarked frames to remove temporally uncorrelated watermarks. Recently, it has been shown that motion-compensation can be used to align similar areas in the neighboring frames before temporal filtering to increase the performance of FTF attack. The resulting attack is known as *motion-compensated frame temporal filtering* (MC-FTF). Subsequently, a new class of video watermarking schemes, referred to as *motion-coherent* (MC) watermarking, has been proposed to counter the MC-FTF attack. In such a scheme, the watermark is coherent with motion in the host frames i.e. pixels in different frames along the motion trajectory carry similar watermark samples. The thesis focuses on the MC-FTF attack and the MC watermarking.

The existing implementations of the MC-FTF attack remove the watermark only from the background regions of the video frames and rely on computationally expensive video mosaicing techniques. We propose a new MC-FTF attack which can remove watermarks from both the background and the foreground. The attack is based on the *motion-compensated redundant temporal wavelet transform* (MC-RTWT) of the watermarked frames. The lifting-based MC-RTWT is applied to the video frames in a scene and the resulting low-pass temporal frames constitute the attacked video. A lifting scheme with adaptive update step is used to improve the visual quality of the attacked video. Experimental results show the effectiveness of the proposed attack in removing the watermark while maintaining a good visual quality of the attacked video.

Since video is generally stored in a compressed format, there is a need for compressed-domain MC-watermarking techniques. We propose two such techniques: one for the MPEG-2 coded video and the other for the sequences coded using the emerging MC-TWT based coding. In the existing MPEG-domain watermarking schemes, a drift compensation signal is added to the inter-coded frames to prevent temporal error propagation due to watermark addition. We have shown that instead of cancelling the drift signal, its proper usage will generate motion-coherent watermarks. We propose a *drift-aided* watermarking scheme for the MPEG-2 coded video. In this scheme, watermarks are added only to the intra-coded frames; MC watermarks for the inter-coded frames are generated during decompression. In the second scheme, the watermarks are added to the low-pass temporal frames obtained from the partial

decompression of sequences coded using the MC-TWT based technique. These watermarks propagate to other frames according to the motion during the decompression and generate MC watermarks for the entire sequence. Experimental results show that the proposed watermarking schemes are robust to the MC-FTF and other inter-frame collusion attacks. It is also shown that motion-coherency in the watermark is a sufficient condition to achieve resistance to known inter-frame collusion attacks.

For a given watermarking system, there exists no simple tool to assess whether the produced watermark is motion-coherent or not. We propose an oracle that reports whether a video sequence contains any motion-incoherent (MIC) component. The oracle is based on the statistical analysis of the motion-compensated prediction error frames (PEF) of the watermarked video. It is observed that there is a clear difference in the distributions of the estimated local variances of the PEFs of the sequences with and without MIC watermarks. This difference in the distributions is exploited to detect the presence of MIC watermarks. Experimental results on watermarked sequences coded at different bit rates clearly demonstrate that the oracle is able to detect the presence of MIC watermarks with good accuracy.

Contents

List of Figures

List of Tables

Nomenclature

1-D	One-dimensional
2-D	Two-dimensional
3-D	Three-dimensional
3D-DWT	Three-Dimensional Discrete Wavelet Transform
A/D	Analogue-to-Digital
D/A	Digital-to-Analogue
DFD	Displaced Frame Difference
DFT	Discrete Fourier Transform
DVD	Digital Versatile Disk
DWT	Discrete Wavelet Transform
FSBM	Fixed Size Block Matching
FTF	Frame Temporal Filtering
FTFR	Frame Temporal Filtering After Registration
GOP	Group of Pictures
iid	Independent and Identically Distributed
kNN	k-Nearest Neighbor
LSB	Least Significant Bit

MAD	Mean Absolute Difference
Mbps	Mega Bits Per Second
MC	Motion-Coherent
MC-FTF	Motion-Compensated Frame Temporal Filtering
MC-RTWT	Hierarchical Variable Size Block Matching
MC-TFA	Motion-Compensated Temporal Frame Averaging
MC-TWT	Motion-Compensated Temporal Wavelet Transform
MIC	Motion-Incoherent
ML	Maximum Likelihood
MM	Method of Moments
MPEG	Moving Picture Expert Group
MSE	Mean Squared Error
NC	Normalized Correlation
PDF	Probability Density Function
PEF	Prediction Error Frame
PSNR	Peak Signal to Noise Ratio
SLIDE	Spatially Localized Image Dependent
SPOMF	Symmetrical Phase Only Filtering
SS	Spread Spectrum
TFA	Temporal Frame Averaging
TWT	Temporal Wavelet Transform
VLC	Variable Length Code
WER	Watermark Estimation and Remodulation

Mathematical Notations

Λ	Two-dimensional grid of points representing the pixel locations in a frame.
$\mathbf{n}$	Pixel location.
$\mathbf{x}_k$	kth host frame.
$\mathbf{w}_k$	Watermark added to the kth host frame.
$\mathbf{y}_k$	kth watermarked frame.
$\hat{\mathbf{y}}_k$	kth attacked frame.
N_f	Number of frames in a sequence.
$\mathcal{F}(\cdot)$	Temporal low-pass filter.
α	Embedding strength of the watermark.
ρ	Correlation coefficient.
$\mathbf{y}_i^{(k)}$	ith watermarked frame after registration with the kth frame.
$\mathcal{M}$	Motion-compensated mapping operator.
$\mathbf{l}_k^i$	kth low-pass frame resulting from the ith level TWT decomposition.
$\mathbf{h}_k^i$	kth high-pass frame resulting from the ith level TWT decomposition.
$\mathbf{c}_k^i$	kth low-pass frame resulting from the ith level MC-RTWT decomposition.
$\mathbf{d}_k^i$	kth high-pass frame resulting from the ith level MC-RTWT decomposition.
$\mathbf{m}$	Motion vector.

Chapter 1

Introduction

The last decades witnessed a transition from the *analogue world* to the *digital world*. The analogue audio and video devices have been replaced with their digital successors. Further, the phenomenal growth of the Internet made the distribution of multimedia data much easier. Although this allows a greater flexibility to the content owners to distribute their valuable contents, security issues like the copyright protection and copy protection have become an issue. In fact, the lack of adequate security measures was one of the reasons behind the delayed introduction of Digital Versatile Disk (DVD) [Rup96].

Digital watermarking was introduced in the mid-90s as a possible solution to the challenging issue of multimedia security. The conventional security measures relied on encryption techniques which offer security only during the transmission. The encryption process applies a mathematical transformation, determined by a secret key, to make the media unintelligible [Sta06]. The intended receiver can invert the transformation through the knowledge of the key that is transmitted through a secure channel. Once the content is decrypted, it can be freely distributed. The basic idea behind digital watermarking is to embed some information into the digital media in an imperceptible way. The embedded information will remain in the content as long as there is considerable degradation to the content. It should be noted that the mechanism of watermarking is complementary to that of encryption.

In the initial period, digital watermarking research was mainly concentrated on still images. Other media like the audio, video, and text are gradually investigated. By the beginning of this century, digital watermarking has become an active area of research. This increase in the interest is evident from the volume of the literature in the area and the increase in the number of conferences and journals.

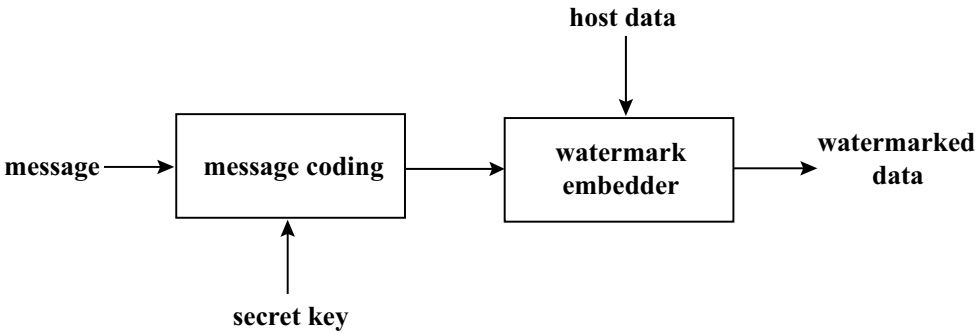


Figure 1.1: Generic watermark embedding process.

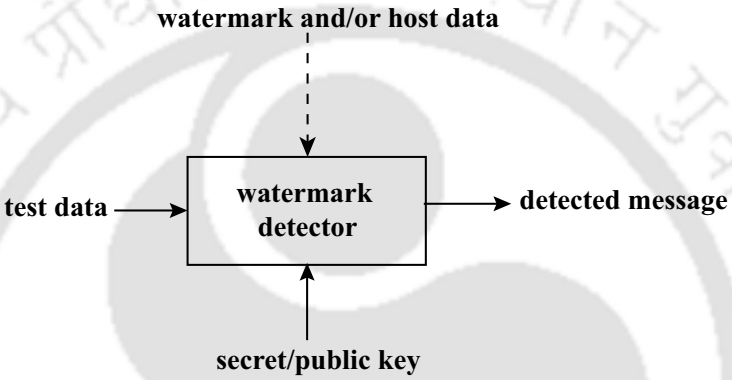


Figure 1.2: Generic watermark detection process.

1.1 Digital Watermarking

All digital watermarking systems consist of two generic processes: *watermark embedding* and *watermark detection*. The block diagram of a generic watermark embedding process is shown in Figure 1.1. The message to be carried by the watermark signal is first coded with a secret key for the security purpose. The coded watermark is then embedded into the host data to generate the watermarked data. The host data may be in the uncompressed or compressed formats. In some cases, the host data may be suitably transformed into another domain before the embedding of the watermark. Figure 1.2 shows the block diagram of a generic watermark detection process. Depending on the availability of the host data and the original watermark, the watermark detection process is classified into three categories. In the *blind* or *oblivious* detection, neither the host nor the watermark is available to the detector. In the case of the *semi-blind* detection, the watermark is available to the detector, but not the host data. If both the host data and the watermark are available to the detector, the detection is *non-blind* or *non-oblivious*. A large number of watermarking algorithms are available in each category [CMB01].

1.1.1 Applications

Digital watermarking, though has emerged as a solution to the copyright and copy protection of digital multimedia data, finds its use in many other applications as well [CM02]. The major applications include:

- **Copyright Protection:** For protecting the intellectual property, a watermark identifying the content owner can be embedded in multimedia data. Such a watermark allows the content owner to prove his/her ownership in the court when someone has infringed on the copyrights.
- **Copy Control:** In the copy control application, the embedded watermark controls the recording device. Depending on the watermark detector output in the recorder, the recorder determines whether the data to be copied or not [BCK$^+$99].
- **Fingerprinting:** In this application, each of the distributed copy contains a unique watermark corresponding to the customer. For example, the watermark may carry a unique identification number corresponding to the customer. Whenever an illegal copy is found, it can be traced back to the customer who has broken the licence agrement, by detecting the embedded watermark [CMB01].
- **Broadcast Monitoring:** The watermark embedded in the commercial advertisements can be used for monitoring whether they are broadcasted as contracted. The watermark can also be embedded in the high valued news items so that an automated broadcast surveillance system can check for any illegal rebroadcasting [KDHM99].
- **Content Authentication:** The authenticity of the data can be checked by embedding a *fragile watermark* which becomes undetectable when the data is altered. Also, by analyzing the detected watermark, the exact location of the alteration can be identified [CMB01].
- **Data Hiding:** In data hiding applications, digital watermarking is exploited for the transmission of some additional information which can be useful in many situations like data retrieval, medical safety and error recovery [Doë05]. Watermarking can also be used in *steganography*: the art of *covert communication*. Recently, it has been revealed that steganogrphic watermarks are used by terrorist organizations like Al-Qaeda to coordinate their activities [GF07].

1.1.2 Trade-Offs

In a watermarking system, there is a complex trade-off between three conflicting parameters: data payload, fidelity and robustness. The balance among these parameters depends on the intended application of the particular watermark.

- **Payload:** The payload of the watermark is the amount of information, in terms of number of bits, carried by the watermark. The payload of the watermark depends on the application. For example, in the copy control applications, a data payload of two bits can encode three messages: *copy-never, copy-once and copy-always*. The copyright protection applications require information about the owner, the year of copyright and the permissions granted. This roughly requires $60 - 70$ bits of information to be embedded in the host data [FG99, KP99]. For the fingerprinting applications, the payload depends on the number of customers. For applications like cable TV and DVD distribution, the potential customers can be as many as $10 \sim 100$ million [HW06].

- **Fidelity:** The fidelity of the watermark refers to the perceptual similarity between the watermarked and the host data [CMB01]. The distortion introduced by the watermark-embedding process should be minimized so that a good fidelity is maintained. That is, a human observer should not be able to notice the changes due to watermark embedding as long as the data is not compared with the host data.

- **Robustness:** The robustness of the watermarking system is the ability of the watermark detector to extract the watermark after the watermarked data is subjected to some modifications, even if its perceptual quality is degraded . The modifications may be an unintentional or an intentional attack to remove the watermark. The common unintentional modifications include signal processing operations like digital-to-analogue (D/A) and analogue-to-digital (A/D) conversions, filtering, compression and noise addition. There are various intentional attacks aimed at removing the watermark, which are discussed in detail in the later part of this Chapter. It should be noted that the required robustness of the watermarking scheme is application dependent. For example, in copyright protection or copy control applications, the watermarking scheme should possess good degree of robustness. However, the fragile watermarks used in the content authentication should not be robust, i.e., the watermark should not survive any modification to the watermarked data.

If the embedding strength of the watermark is increased to increase the robustness, the fidelity of the watermark is compromised. Similarly, an increase in the payload of the watermark results in the reduced robustness. Thus, the three parameters of payload, fidelity and robustness are of conflicting nature.

1.2 Video Watermarking

In the early stages of video watermarking research, the main trend was to re-use the techniques developed for the more mature still image watermarking. In these approaches, the video is considered as a sequence of still images and image watermarking is applied to each frame independently [DD03a]. The additional temporal domain offers a larger signal space for video watermarking. This could be effectively exploited for increasing the payload or reducing the visual impact. One important difference between the still image watermarking and video watermarking is that the latter often imposes real-time or near real-time constraints on the watermarking system [HK99].

There are many video specific watermarking schemes proposed in literature. These schemes can be broadly categorized into two groups. The first category consists of the watermarking schemes developed for uncompressed or raw video sequences whereas the watermarking schemes developed for the compressed sequences belong to the second category. The following subsections describe some of the major works in each category.

1.2.1 Watermarking of Uncompressed Video

In [HG98], Hartung et al. proposed one of the pioneering methods for watermarking of uncompressed video. The video sequence to be watermarked is first converted to a 1-D signal by scanning the frames in a line-by-line manner. The watermark message, spread with a given chirp-rate and modulated with a binary pseudo-noise sequence is then added to the line-scanned video signal. Finally, the resulting signal is rearranged to obtain the watermarked video sequence. The detection process employs a simple correlation detector. The watermarked sequence is line-scanned in the same fashion as in the embedder and then demodulated with the pseudo-random noise sequence. For each of the watermark message bit, the demodulated signal is summed over the corresponding spreading window and the sign of the summation gives the decoded bit. Since the host video is not used in the detection process, the correlator output may be affected by the cross-talk between the host and the watermark. The authors of [HG98]

propose to minimize the cross-talk by spatial high-pass filtering of the watermarked sequence before computing the correlation.

A video watermarking system for broadcast monitoring, called the JAWS (Just Another Watermarking System), has been proposed by the researchers from the Philips corporation [KDHM99]. This scheme embeds a fixed watermark pattern in the consecutive frames of the video. The basic building block of the watermark is a *reference pattern* of size 128×128 pixels, drawn from a white Gaussian random process using a secret key. For each bit of the hidden message, a reference watermark is generated as the difference between the reference pattern and its cyclically shifted version according to the message bit. The reference watermark is then tiled to generate the watermark for the whole video frame. In order to minimize the visual distortion, the watermark to be embedded in each frame is perceptually shaped according to a local activity measure and computed using a spatial high-pass Laplacian filter. Finally, the perceptually shaped watermark is scaled with a global scaling factor and added to the host frames to generate the watermarked sequence. The watermark detector is a correlation detector. Also, a pre-filtering step is used in the detector to reduce the cross-talk between the host and the watermark frames. The detector is designed so that any spatial shift in the watermark frame does not affect the watermark detectability. This is achieved by using the symmetrical phase only filtering (SPOMF), a detection method originally proposed for pattern recognition. In a subsequent work [TSV+00], the watermark detector is modified to achieve robustness to scaling.

Another approach in the watermarking of uncompressed sequences is to consider the video as a 3-D signal and embed watermark in the 3-D transform domain. Many 3-D transforms like the Discrete Fourier Transform (DFT), the Gabor transform and the wavelet transform have been investigated [DCRP99, ZWH04, CN05]. This approach is motivated by the improved visual quality and the increased robustness. However, these advantages are counterbalanced by the increased computational and memory requirements.

1.2.2 Watermarking of Compressed Video

The main advantage of watermarking the compressed video is that the watermark embedder can process the compressed stream which has a much lower data rate than the uncompressed video. Further more, such an approach does not require the re-compression which is computationally demanding. The state-of-the-art video coding techniques use a combination of the predictive coding and the transform coding to exploit the spatio-temporal redundancies present in the video sequences. The coded stream consists

of the coded motion vectors, and the transform coefficients which are coded using the variable length codes (VLCs). Depending on the modification to the coded streams, the compressed-domain watermarking schemes can be classified into three groups. In the first approach, the watermark embedding algorithms modify the motion vectors in the compressed stream. The second group, the watermark embedding is done directly in the VLCs. The third and the most popular approach is to embed the watermark in the transformed coefficients obtained by decoding the VLCs. Note that there are algorithms in which the coded sequence is completely decompressed prior to the watermark embedding [DSR98]. These techniques, even though operate on the compressed video, cannot be classified as compressed-domain watermarking.

In [JKE97], a watermarking scheme that embeds the information into the motion vectors has been proposed. The motion vectors are pseudo-randomly quantized to enforce a parity rule. For example, the horizontal component of the motion vector is quantized to an even value if the message bit to be embedded is 0, and to an odd vale if the bit is 1. For visibility constraints, only the motion vectors from the smooth areas are altered. The embedded watermark can be retrieved directly from the motion vectors of the watermarked stream. However, if the watermarked sequence is decompressed, it has to be recompressed for retrieving the watermark. Other watermarking schemes operating on the motion vectors have been proposed [ZLZ01, BLD03]. The main drawback of this approach is that, it is difficult to predict the impact of modifying the motion vectors on the perceptual quality of the watermarked sequence. As a result, this approach, even though quite simple, has received little attention.

Langelaar [Lan00] develops a watermarking technique by modifying the VLC codewords in the MPEG stream. In this scheme, the least significant bits (LSB) of selected VLCs are changed in such a way that their quantized level is equal to the watermark message bit. The VLCs for watermark embedding are chosen such that the perceptual quality of the sequence is not affected and the MPEG stream keeps its original size. The main advantage of this scheme is its low computational requirements which is highly desirable in real-time applications. However, the drawback is that the watermark embedding and detection are completely dependent on the structure of the compressed stream. So this watermarking scheme is not robust to transcoding, i.e., the watermark can be removed by decompressing the stream and then recompressing it at a different bit-rate.

In [HG98], Hartung and Girod propose a spread-spectrum watermarking scheme for MPEG-2 coded video. A pseudo-noise signal is generated with a secret key and modulated by the message bits to generate the watermark. The watermark is then 8×8 block DCT transformed and added to the non-zero DCT coefficients obtained by the partial decoding of the MPEG-2 coded host sequence. A

rate control mechanism ensures that the bit-rate does not increase as a result of watermark embedding. In the watermark detector, the sequence is first decompressed and then a correlation-based detection method is employed. Another compressed-domain watermarking scheme, called the DEW (differential energy watermarking), has been proposed in [LL01]. The scheme is based on enforcing an energy difference on the high frequency DCT coefficients on selected blocks. For each bit of the message to be embedded, a set of n 8×8 blocks are pseudo-randomly chosen from the video frame and then divided into two subsets of equal size. A typical value of n varies between 16 and 64. The energy of the high frequency DCT coefficients in one or other subset is reduced such that the sign of the energy difference is equal to the message bit. The energy is reduced by discarding the high frequency DCT coefficients below a cut-off point in the subgroup obtained by a zig-zag scanning. The watermark message bit can be retrieved by simply measuring the energy difference between the DCT coefficients in the subgroups, selected in the same way as in the encoder. Though the scheme was originally proposed for intracoded frames in the MPEG-2 stream, it is later extended for the inter-coded frames as well [SL01].

Most of the proposed compressed domain watermarking schemes are specifically designed for the sequences coded using MPEG-2, the most widely used compression standard. Some watermarking schemes have been proposed for sequences coded using more advanced compression standards like the MPEG-4 [HEG98, ALC03, BBC05] and the H.264 [NM07, ZHQM07].

1.3 Attacks on Watermarking Systems

A digital watermarking system when deployed in security related applications like copy control and fingerprinting, is likely to undergo many hostile attacks. A lot of research effort has been devoted to benchmarking and improving the robustness against common signal processing operations like filtering, lossy compression and noise addition. However, only a few works have evaluated the impact of malicious intelligence, i.e., the ability to survive the attacks of a hostile adversary on the watermarking system. Within the watermarking community, this is regarded as an additional specification, known as the *security*, to highlight the difference with robustness requirements, which deal with non-hostile data manipulation [DD05]. The first attempt to make a clear distinction between the watermark security and the robustness was by Kalker [Kal01] with the following definitions:

"Robust watermarking is a mechanism to create a communication channel that is multiplexed into the host data such that the perceptual degradation of the marked data with the original data is minimal and the capacity of the channel degrades as a smooth function as the degradation of the marked content".

"Watermark security refers to the inability of unauthorised users to either (i) remove, (ii) detect and estimate, (iii) write, or (iv) modify the original watermark message"".

Security evaluation has recently received an increasing interest in the watermarking research community [BBF03, Fur05, PTW06]. In particular, theoretical studies have been conducted to measure how much information about the watermark can be gained by an attacker [CFF05, PFCPG05, CPFPG05]. The increasing interest on this topic is evident from the number of special sessions organized on these topics in recent conferences [BBF02, BPG05a, BPG05b, PGF05].

1.3.1 Attack Classification

There are many different classifications of the watermarking attacks proposed in literature [PAK98, VPP+01, Kal01, DD05]. Following [DD05], the attacks can be broadly classified into two groups: *attacks on robustness* and *attacks on security*. The main properties that distinguish these two attack categories are the *intention* of the attack and the knowledge about the watermarking system that the attacker exploits [DD05, Fur05]. The robustness attacks are common signal processing operations like lossy compression and denoising which are generally not intended to remove the watermark. The security attacks on the other hand involve pirates whose intention is to defeat the watermarking system. The robustness attacks are blind in the sense that the attacker does not use any knowledge about the watermarking system. But in the security attack, the attacker gains some knowledge about the watermarking system that could be effectively exploited to defeat it.

The attacks on robustness can be further divided into two groups: *synchronous* and *asynchronous* attacks. Synchronous attacks include common signal processing operations like A/D and D/A conversions, filtering, lossy compression and noise addition. The asynchronous or geometric attack is accomplished by geometrically transforming the watermarked data. Since each point in the watermarked data is associated with a given bit of the watermark, any geometrical transformation results in desynchronization in the watermark detector and renders the watermark undetectable. The spread-spectrum watermarking schemes which employ a correlation detector is particularly vulnerable against this attack. In addition to the spatial transformations, the video watermarking schemes may be affected by temporal desynchronization operations like frame-rate changing, frame dropping, frame swapping and frame insertion [LD04].

Depending on the knowledge about the watermarking system that the attacker exploits, the security attacks can be divided into *cryptographic* and *protocol* attacks. In the cryptographic attacks, the

attacker gains some knowledge about the embedded watermark prior to the attack. Main attacks in this category include the brute force search, the oracle attack, the copy attack and the collusion attack. In the brute force search method, the attacker aims to find the secret key used in the watermark embedder by trying all the possible keys. The basic idea behind the oracle attack is to iteratively modify the watermarked data with the help of publicly available watermark detectors, until the watermark becomes undetectable. In the copy attack, the attacker estimates the watermark and then inserts it into another unwatermarked content. The collusion attack, also known as the statistical attack, gathers different watermarked contents and combines them to remove the embedded watermarks. This attack is further detailed in the following section. The protocol attack exploits some general knowledge about the watermarking system and the main attacks in this category are the *deadlock attack* and the *mosaic attack*. The deadlock attack is concerned with the copyright protection applications of watermarking. If an attacker embeds his own watermark in a watermarked content, the watermarking detectors cannot determine which watermark was added first and this leads to a deadlock regarding the the ownership of the content. The mosaic attack is aimed at automated search engines which download the images from the Internet and check weather they contain any watermark [PAK98]. The watermarked image is divided into sub-images such that the watermark is not detectable from the sub-images. These sub-images are stored in a suitable sequence in the web page. When the image is rendered, the sub-images stuck back together.

1.4 Collusion Issues in Video Watermarking

Among the attacks on watermarking systems, the one that has received probably the most attention is the collusion attack. In the collusion attack, a set of malicious users combine their watermarked contents to obtain a watermark-free copy [DD03a]. The collusion attack was first investigated in the case of still images and two types of collusion attacks have been identified. The first type of collusion attack is possible when the same watermark is embedded into different copies of different contents, for example, the watermarks for copyright protection. An estimate of the watermark can be obtained from each of the watermarked copies and the individual estimates are then combined to obtain a refined estimate [VPH$^+$00] of the watermark. This refined estimate can be used to remove the watermark. The second type of collusion is possible in the fingerprinting applications where different copies of the same content carry different watermarks. From a linear or nonlinear combination of these copies, it is possible to obtain an unwatermarked copy of the content [ZWWL05].

The video watermarking is vulnerable to two types of collusion: *inter-video* and *inter-frame* or *intra-video* collusion. In inter-video collusion, copies of the same video with different watermarks or different videos with the same watermark are combined. On the other hand, in inter-frame collusion, an attacker collects a number of frames from a single video sequence combines them to remove the embedded watermark. The inter-video collusion requires the collaboration of a group of attackers whereas the inter-frame collusion requires only a single copy of the video.

Two approaches have been proposed to counter the collusion attacks in fingerprinting applications. The first approach uses *collusion-resistant fingerprints* [BS98, WTWL04]. The watermarks embedded into different copies of the same multimedia content are designed such that it is possible to identify at least one of the attackers from the colluded copy. In the second approach, the watermarked copies distributed to different customers are *intensionally desynchronised* so that the perceptual quality of the colluded copy is significantly degraded [CST04, MM05].

The danger of inter-frame collusion in video watermarking was first addressed by Swanson *et al.* [SZT98]. They proposed a watermarking scheme based on the multi-resolution temporal decomposition of the video. The watermark generated by this scheme consists of *temporally static* and *dynamic* components. Kundur et al. [SKH02] further investigated the problem and proposed an embedding rule for inter-frame collusion-resistant watermarking. According to this embedding rule, the watermark embedded into a pair of frames should be as correlated as the corresponding host frames. In a recent work, Doerr *et al.* [DD04a] showed that this embedding rule is not enough to guarantee the robustness to inter-frame collusion. They have shown that the motion information in the video sequences can be exploited to devise more effective inter-frame collusion attacks. They proposed a new inter-frame collusion attack, the *frame temporal filtering after registration* (FTFR) to remove uncorrelated watermark samples embedded along the motion trajectories [DD03b]. Subsequently, a novel video watermarking strategy, referred to as *motion-coherent watermarking*, has been proposed to counter the FTFR attack [DD04a]. In this watermarking scheme, the watermark is *coherent* with motion in the host frames and hence possess a good degree of robustness against the FTFR attack.

1.5 Motivation of the Present Work

The inter-frame collusion is a powerful attack against most of the existing video watermarking schemes. The real danger of the attack is that unlike the inter-video collusion; it requires only a single copy of the watermarked sequence to perform the attack. However, only few works have investigated the inter-

frame collusion and its countermeasures. In particular, motion-coherent watermarking remains largely an unexplored area. The objective of the thesis is to investigate how the motion information in the video sequence can be exploited to improve inter-frame collusion and design motion-coherent watermarking to counter such collusion attacks. Keeping in view this objective, the thesis proposes a new inter-frame collusion attack, two motion-coherent watermarking schemes and a computationally efficient tool to assess whether a given video sequence contains any motion-incoherent watermark.

1.6 Outline of the Thesis

The organization of the rest of the thesis is as follows:

In Chapter 2, we investigate how to improve the performance of the inter-frame collusion attack by exploiting the motion information in the video frames. The chapter proposes a new *motion-compensated frame temporal filtering attack*. Detailed analytical and experimental results are presented to demonstrate the effectiveness of the proposed attack.

Chapter 3 focuses on developing computationally efficient motion-coherent watermarking schemes for compressed videos. The chapter first reviews the existing watermarking schemes which uses the motion information, including the motion-coherent watermarking schemes. The inter-frame collusion resistance properties of the existing compressed-domain watermarking schemes are also analyzed. The chapter proposes two compressed-domain motion-coherent watermarking schemes. A detailed analysis of some important features of the motion-coherent watermarking schemes is presented.

Chapter 4 presents a novel tool for assessing the motion-coherency in the watermark. We propose a simple *oracle*, which accurately reports whether a given watermarked sequence contains any motion-incoherent watermark component or not.

Chapter 5 summarizes the main contributions of the thesis and suggests a few tracks for further investigation.

Chapter 2

Motion-Compensated Inter-Frame Collusion

In the early developments of video watermarking, the research community did not pay much attention to the importance of the temporal dimension in the video. As a result, the inter-fame collusion attack was evolved as a threat to the video watermarking schemes. The inter-frame collusion attacks have been shown to be very powerful to defeat the most commonly used video watermarking schemes, in particular the frame-by-frame watermarking strategies. Subsequently, many watermarking schemes have been proposed to counter the inter-frame collusion. Theoretical studies were also conducted to define the required properties of the watermark to counter these attacks. However, these studies ignored another important and distinguishing feature of the video sequences: *the motion information*. Lately, it has been shown that the motion information in the sequences can be exploited to design a *motion-compensated inter-frame collusion attack*. This attack is more effective against many watermarking schemes including the existing inter-frame collusion-resistant ones. This chapter investigates the motion-compensated inter-frame collusion and proposes one such attack. The chapter first reviews the basic inter-frame collusion attacks and their countermeasures. The motion-compensated inter-frame collusion attack is then investigated. Finally, the proposed attack is presented along with the theoretical analysis and experimental results.

2.1 Inter-Frame Collusion

2.1.1 Basic Attacks

The basic idea behind the inter-frame collusion attack is the exploitation of the redundancy, either in the host video frames or in the embedded watermark, to estimate the redundant component. Depending

on the redundancy, two types of inter-frame collusion attacks are possible : *Type I* and *Type II* collusion [DD03a].

Type I: Watermark Estimation

Due to the fidelity constraint, most watermarking schemes embed the watermark in the spatial high-frequency components of the host video signal. As a result, for each video frame, it is possible to obtain a rough estimate of the underlying watermark in each frame. For example, by computing the difference between a watermarked frame and its low-pass filtered version, one can obtain a rough estimate of the watermark. Each individual estimate is not accurate enough to compromise the performances of the detector. However, if the watermark is temporally redundant, it is possible to obtain a refined estimation of the watermark by combining these uncorrelated individual approximations [HMY00, SKH02]. Re-modulating this estimated watermark is then usually enough to defeat the detector. This baseline attack is referred to as *watermark estimation and remodulation* (WER) [VPH$^+$00]. In this case, the temporal redundancy of the watermark signal is exploited to acquire some knowledge about it from several uncorrelated observations. This strategy is most effective when correlated watermarks are embedded within visually dissimilar frames, e.g. the key-frames from a video. It should be noted that this baseline attack can be extended to more complex watermarking strategies. For instance, it is possible to estimate a finite set of watermark patterns or even a low-dimensional watermarking subspace [DD04b]. Recent theoretical studies have added more support to these empirical results. Using information-theoretic tools, they quantify how much information about the watermark leaks through several observations and how many observations are required to defeat the system [CFF05, PFCPG05, CPFPG05].

Type II: Host Signal Estimation

In contrast with the previous attacks, the idea here is to exploit the temporal redundancy of the host video frames. In other words, this attack is relevant when visually similar watermarked frames are available, e.g. successive frames from the same sequence. Indeed, if successive video frames are carrying uncorrelated watermarks, it is possible to estimate the original host video frames by applying a low-pass temporal filter [DCP00, SKH02]. This attack is generally known as *frame temporal filtering* (FTF) attack [DD03b] which is mathematically presented below.

Consider a sequence of N_f frames each of size $N_1 \times N_2$. Suppose $\Lambda = \{(n_1, n_2)| \quad (n_1, n_2) \in \mathbb{Z}^2, 0 \leq n_1 \leq N_1 - 1, 0 \leq n_2 \leq N_2 - 1\}$ be a two-dimensional grid of points representing the positions

of pixels in the frame and $\{\mathbf{y}_k[\mathbf{n}],\ k = 0, 1, \cdots, N_f - 1$ and $\mathbf{n} \in \Lambda\}$ be a sequence of watermarked video. The FTF attack is given as

$$\hat{\mathbf{y}}_k[\mathbf{n}] = \mathcal{F}(\mathbf{Y}_k), \qquad \mathbf{Y}_k = \{\mathbf{y}_i[\mathbf{n}], 0 \leq \mid i - k \mid < L/2\} \tag{2.1.1}$$

where $\mathcal{F}(\cdot)$ is a temporal low-pass filter, L is the width of the filtering window and $\hat{\mathbf{y}}_k[\mathbf{n}]$ is the kth attacked frame. In the case of the simple frame averaging attack with an odd temporal window width, we get

$$\hat{\mathbf{y}}_k[\mathbf{n}] = \frac{1}{L}\sum_i \mathbf{y}_i[\mathbf{n}], \qquad 0 \leq \mid i - k \mid < L/2 \ . \tag{2.1.2}$$

Let $\{\mathbf{w}_k[\mathbf{n}],\ k = 0, 1, \cdots, N_f - 1$ and $\mathbf{n} \in \Lambda\}$ be the sequence of frame-by-frame additive watermarks. Further, assume that each watermark frame $\mathbf{w}_k[\mathbf{n}]$ consists of *independent and identically distributed* (iid) Gaussian random variables with zero mean and unit variance. Then the watermarked frame $\mathbf{y}_k[\mathbf{n}]$ can be represented as

$$\mathbf{y}_k[\mathbf{n}] = \mathbf{x}_k[\mathbf{n}] + \alpha\mathbf{w}_k[\mathbf{n}], \qquad \mathbf{w}_k[\mathbf{n}] \sim \text{iid}\, \mathcal{N}(0, 1), \quad k = 0, \cdots, N_f - 1 \tag{2.1.3}$$

where α is a constant scaling factor, known as the *embedding strength*. Substituting $\mathbf{y}_k[\mathbf{n}]$ from Equation (2.1.3) in Equation (2.1.2) results

$$\hat{\mathbf{y}}_k[\mathbf{n}] = \frac{1}{L}\sum_i \mathbf{x}_i[\mathbf{n}] + \alpha\frac{1}{L}\sum_i \mathbf{w}_i[\mathbf{n}] \ . \tag{2.1.4}$$

If the watermarks embedded in different frames are statistically independent of each other and L is large, then (2.1.4) can be rewritten as,

$$\hat{\mathbf{y}}_k[\mathbf{n}] \approx \frac{1}{L}\sum_i \mathbf{x}_i[\mathbf{n}] \ . \tag{2.1.5}$$

The width of the temporal window used for filtering is limited by the dynamic content of the video sequence. In *static scenes*, large window widths can be used without degrading the visual quality of the attacked video. If the video frames contain camera motion and/or moving objects, severe blurring and ghosting artifacts are likely to occur. Hence a lower window width should be used to preserve the visual quality of the attacked video. However, this also reduces the efficiency of the attack. The FTF attack is more effective in frames from a *static scene* carrying uncorrelated watermarks.

2.1.2 Countermeasures to Basic Attacks

The pioneering work in collusion-resistant video watermarking was done by Swanson *et al.* [SZT98]. They argued that a watermarking scheme is collusion-resistant only if the embedded watermark is

statistically invisible. Based on this principle, they proposed a collusion-resistant video watermarking scheme using the *temporal wavelet transform* (TWT). The video frames are segmented into scenes and the TWT is applied to the frames from each scene. The watermark is added to the temporal wavelet frames using the frequency masking and spatial masking properties of the human visual system to increase the robustness of the watermark. The inverse temporal wavelet transform is then applied to get the watermarked frames. The watermark is a combination of temporally *static* and *dynamic* components. The portion of the watermark embedded in the temporal low-pass frame exists in all the frames in a scene, whereas that embedded in the high-pass frames is temporally localized. The *dynamic* part of the watermark will prevent Type I collusion and the *static* part will survive Type II collusion.

The correlation between two frames is a measure of the statistical similarity between the frames. Based on this fact, Kundur *et al.* [SKH05a] proposed a basic embedding rule for collusion-resistant video watermarking given as

$$\rho(\mathbf{w}_i[\mathbf{n}], \mathbf{w}_j[\mathbf{n}]) \approx \rho(\mathbf{x}_i[\mathbf{n}], \mathbf{x}_j[\mathbf{n}]), \qquad \forall (i,j) \in \{0, 1, \cdots, N_f - 1\} \tag{2.1.6}$$

where ρ is the correlation coefficient, $\mathbf{w}_i$ is the watermark embedded in the i^{th} host frame, $\mathbf{x}_i[\mathbf{n}]$. In other words, the embedded watermarks should be as correlated as the corresponding host video frames. Based on this embedding rule, Kundur *et al.* [SKH05b] proposed the *spatially localized image dependant* (SLIDE) video watermarking scheme. In this scheme, the watermark is embedded in spatially localized sub-frames centered around a set of image dependent anchor points. The anchor points are selected in such a way that, given two frames, the cardinality of the intersection set of the anchor points is directly proportional to the correlation between the frames. As a consequence, the pair-wise correlation between the watermarks will be proportional to that between the host frames and the watermarking scheme achieves the inter-frame collusion-resistance.

2.1.3 Frame Temporal Filtering After Registration (FTFR)

It is clear from the previous discussion on the FTF attack that (a) when the video frames contain moving objects, the FTF attack will not succeed without severe degradation in the quality of the attacked video and (b) the FTF attack will not be successful if the watermarks embedded in the frames are highly correlated (e.g. a fixed watermark in all the video frames). A more effective FTF attack is possible by exploiting the motion in the video frames. Such an attack, called the *frame temporal filtering after registration* (FTFR), has been proposed by Doërr *et al.* [DD03b]. The basic idea behind this attack is to compensate camera motion before the temporal filtering. Each watermarked frame is registered with

a reference frame before the temporal filtering. The attacked frame $\hat{\mathbf{y}}_k[\mathbf{n}]$ is, therefore, given by

$$\hat{\mathbf{y}}_k[\mathbf{n}] = \mathcal{F}(\dot{\mathbf{Y}}_k) \tag{2.1.7}$$

where $\dot{\mathbf{Y}}_k = \{\mathbf{y}_i^{(k)}[\mathbf{n}], 0 \leq \mid i-k \mid \leq L/2\}$ and $\mathbf{y}_i^{(k)}[\mathbf{n}]$ is the ith watermarked frame after registration with the kth frame. The reported experimental results show that embedding neither a fixed watermark nor an uncorrelated watermark in each frame will survive the FTFR attack (except for the frames from a static scene carrying the same watermark in all the frames).

In addition to the camera motion, the motion of the objects also needs to be considered for frame registration. Using a complex motion model, Doërr *et al.* [DD04a] extended their previous work to a more effective collusion attack. In this attack, the background of a given frame is replaced with one estimated from the neighboring frames. First, the moving objects in the given frame and the neighboring frames are separated from the background using a video object segmentation technique. The resulting background frames are registered with a reference frame and averaged to get an estimate of the background of the target frame. For registration of the background frames, a first-order polynomial motion model is used. This model involves the *zoom factor*, 2*-D rotation angle*, 2*-D translation* and the *co-ordinates of the optical center* of the camera. Prior to registration, the model parameters are estimated for each frame. Finally the objects in the target frame are inserted back into the estimated background to get the attacked frame. To counter this FTFR attack, they have proposed a background watermarking scheme in which the camera motion is compensated before embedding the watermark.

Motion in video sequences is due to both the camera motion and the motion of the objects. In the FTFR attack, only the camera motion is exploited. The portion of the watermark embedded in the moving objects will not be affected by the attack. So, the FTFR attack is less effective when moving objects occupy a considerable part of the video scene. Another drawback is its high computational complexity particularly when the video scene contains complex motion or multiple objects.

2.2 Proposed Attack

In pursuit to develop a more effective FTF attack, we propose to apply temporal filtering by using the *motion-compensated redundant temporal wavelet transform* (MC-RTWT). The MC-RTWT is a special case of the motion-compensated temporal wavelet transform (MC-TWT), which is used in video coding techniques [OdSW04]. The following subsections first explain the motivation in choosing the MC-RTWT and then present the proposed attack.

2.2.1 Motion-Compensated Temporal Wavelet Transform

The three-dimensional wavelet coding of video has been an active area of research as an alternative to the conventional hybrid coding techniques [CW99]. In this approach, the transform coding is extended to the temporal direction by applying the discrete wavelet transform (DWT) along the temporal axis. In the three-dimensional discrete wavelet transform (3D-DWT) based scalable video coding techniques, the low-pass temporal wavelet frames are used to represent the reduced frame-rate video. If the video scene contains moving objects, it will introduce ghosting artifacts into the temporal low-pass frames and substantial energy coefficients into the high-pass frames. This results in the reduction of the coding efficiency and the visual quality of the reduced frame-rate video. This is because of the fact that motion in the video frames is not considered during the temporal filtering. These drawbacks can be reduced if the TWT is performed along the motion trajectories [ST03]. The resulting transform is known as the *motion-compensated temporal wavelet transform* (MC-TWT).

The MC-TWT can be implemented using the *transversal* or the *lifting-based* approach [Kon04]. The lifting scheme is an efficient way of implementing the wavelet transform [DS98]. It divides the wavelet transform into a set of *prediction* and *update* steps. In addition to low computational complexity, the lifting-based approach can incorporate any motion model (local or global) with sub-pixel accurate motion [ST03].

Following the notation used in [ST03], suppose $\mathcal{M}_{k\to l}(\mathbf{n})$ denote a point in the frame $\mathbf{y}_k$ which has moved to point $\mathbf{n}$ in the frame $\mathbf{y}_l$. Then

$$\mathcal{M}_{k\to l}(\mathbf{n}) = \mathbf{n} + \Delta\mathbf{n}, \qquad \mathbf{n} \in \Lambda$$

where $\Delta\mathbf{n}$ is the *motion vector* associated with the pixel $\mathbf{n}$ in the frame $\mathbf{y}_l$. We can now write

$$\mathbf{y}_k[\mathcal{M}_{k\to l}(\mathbf{n})] \equiv \mathbf{y}_k[\mathbf{n} + \Delta\mathbf{n}] \approx \mathbf{y}_l[\mathbf{n}], \qquad \mathbf{n} \in \Lambda \ .$$

Depending on the precision of the motion model, the point $\mathcal{M}_{k\to l}(\mathbf{n})$ might not belong to Λ. Spatial interpolation is used to obtain the intensity values of such *sub-pixel* points.

For example, one-level MC-TWT decomposition of a video sequence $\{\mathbf{y}_k[\mathbf{n}]\}$ using the Haar filter can be implemented by the following lifting steps

$$\left.\begin{array}{rl} \text{Prediction step}: & \mathbf{h}_k^1[\mathbf{n}] = \mathbf{y}_{2k+1}[\mathbf{n}] - \mathbf{y}_{2k}[\mathcal{M}_{2k\to 2k+1}(\mathbf{n})] \\ \text{Update step}: & \mathbf{l}_k^1[\mathbf{n}] = \mathbf{y}_{2k}[\mathbf{n}] + \frac{1}{2}\mathbf{h}_k^1[\mathcal{M}_{2k+1\to 2k}(\mathbf{n})] \end{array}\right\}, \qquad k = 0, 1, \cdots, (N_f/2) - 1 \tag{2.2.1}$$

where $\mathbf{h}_k^1[\mathbf{n}]$ and $\mathbf{l}_k^1[\mathbf{n}]$ denote the high-pass and low-pass temporal frames respectively. The corresponding reconstruction lifting steps are given by

$$\left.\begin{aligned}\mathbf{y}_{2k}[\mathbf{n}] &= \mathbf{l}_k^1[\mathbf{n}] - \tfrac{1}{2}\mathbf{h}_k^1[\mathcal{M}_{2k+1\to 2k}(\mathbf{n})] \\ \mathbf{y}_{2k+1}[\mathbf{n}] &= \mathbf{h}_k^1[\mathbf{n}] + \mathbf{y}_{2k}[\mathcal{M}_{2k\to 2k+1}(\mathbf{n})]\end{aligned}\right\}, \qquad k = 0, 1, \cdots, (N_f/2) - 1 \ . \tag{2.2.2}$$

Similarly, the prediction and update lifting steps for the MC-TWT using the 5/3 bi-orthogonal wavelet are given by

$$\left.\begin{aligned}\mathbf{h}_k^1[\mathbf{n}] &= \mathbf{y}_{2k+1}[\mathbf{n}] - \tfrac{1}{2}\big(\mathbf{y}_{2k}[\mathcal{M}_{2k\to 2k+1}(\mathbf{n})] + \mathbf{y}_{2k+2}[\mathcal{M}_{2k+2\to 2k+1}(\mathbf{n})]\big) \\ \mathbf{l}_k^1[\mathbf{n}] &= \mathbf{y}_{2k}[\mathbf{n}] + \tfrac{1}{4}\big(\mathbf{h}_{k-1}^1[\mathcal{M}_{2k-1\to 2k}(\mathbf{n})] + \mathbf{h}_k^1[\mathcal{M}_{2k+1\to 2k}(\mathbf{n})]\big)\end{aligned}\right\}, \quad k = 0, 1, \cdots, (N_f/2)-1 \tag{2.2.3}$$

and the corresponding reconstruction lifting steps are

$$\left.\begin{aligned}\mathbf{y}_{2k}[\mathbf{n}] &= \mathbf{l}_k^1[\mathbf{n}] - \tfrac{1}{4}\big(\mathbf{h}_{k-1}^1[\mathcal{M}_{2k-1\to 2k}(\mathbf{n})] + \mathbf{h}_k[\mathcal{M}_{2k+1\to 2k}(\mathbf{n})]\big) \\ \mathbf{y}_{2k+1}[\mathbf{n}] &= \mathbf{h}_k^1[\mathbf{n}] + \tfrac{1}{2}\big(\mathbf{y}_{2k}[\mathcal{M}_{2k\to 2k+1}(\mathbf{n})] + \mathbf{y}_{2k+2}[\mathcal{M}_{2k+2\to 2k+1}(\mathbf{n})]\big)\end{aligned}\right\}, \quad k = 0, 1, \cdots, (N_f/2)-1 \ . \tag{2.2.4}$$

Note that the MC-TWT with the Haar wavelet uses *unidirectional* motion-compensated prediction where as that with the $5/3$ wavelet uses *bidirectional* motion-compensated prediction.

If the MC-TWT is applied along the *true* motion trajectory, the resulting low-pass temporal frames will have high visual quality. But there are places like scene changes and occluded/uncovered regions where any motion model must necessarily fail. When the motion model fails to follow the true motion trajectory, the energy in the temporal high-pass frames increases and the subsequent update step adds these high energy coefficients back into the temporal low-pass frames. This causes ghosting artifacts in the low-pass temporal frames. Thus, there exists a direct relationship between the ghosting artifacts in the temporal low-pass frames and the energy in the temporal high-pass frames. A method for reducing the ghosting artifacts from the temporal low-pass frames has been proposed in [MT03]. In this method, the update lifting steps are weighted according to the energy in the temporal high-pass frames. The normalized energy in the temporal high-pass frames are mapped to an *update weight* with a decreasing function of energy. In the lifting step with *adaptive update*, the values added to the even-numbered frames are multiplied with these weights.

Motion-compensated Redundant Wavelet Transform

The number of low-pass frames obtained after the MC-TWT decomposition of a sequence depends on the level of decomposition. To obtain the same number of low-pass frames as that of the sequence, the

down-sampling operation in the DWT can be avoided. The resulting *redundant discrete wavelet transform* (RDWT) is an *overcomplete* representation of a signal [She92]. The RDWT is a shift-invariant transform and each resulting subband output is of the same size as that of the input signal. The latter property is exploited in the proposed attack. The two-dimensional RDWT (2D-RDWT) has been used in applications like denoising, edge detection, and 3-D video coding [GKF02, SH97, WCF03]. It has also been used in image watermarking and shown to be more robust to compression attack than its critically sampled counterpart [PF05].

Similar to 2D-RDWT, the MC-TWT introduced in the above subsection can also be made overcomplete by removing the downsampling operation. We denote this as the *motion-compensated redundant temporal wavelet transform* (MC-RTWT). For example, an L-level MC-RTWT decomposition using the Haar filter can be obtained by the following iterative lifting steps:

$$\mathbf{c}_k^0[\mathbf{n}] = \mathbf{y}_k[\mathbf{n}], \qquad k = 0, 1, \cdots, N_f - 1$$

$$\mathbf{d}_k^{i+1}[\mathbf{n}] = \begin{cases} \mathbf{c}_k^i[\mathbf{n}] - \mathbf{c}_{k-2^i}^i[\mathcal{M}_{k-2^i \to k}(\mathbf{n})], & k = 2^i, 2^i+1, \cdots, N_f - 1 \\ \mathbf{c}_{k-2^{i+1}}^i[\mathbf{n}] - \mathbf{c}_{k-2^i}^i[\mathcal{M}_{k-2^i \to k-2^{i+1}}(\mathbf{n})], & k = N_f, N_f+1, \cdots, N_f - 1 + 2^i \end{cases} \tag{2.2.5}$$

$$\mathbf{c}_k^{i+1}[\mathbf{n}] = \mathbf{c}_k^i[\mathbf{n}] + \frac{1}{2}\mathbf{d}_{k+2^i}^{i+1}[\mathcal{M}_{k+2^i \to k}(\mathbf{n})], \qquad k = 0, 1, \cdots, N_f - 1 \tag{2.2.6}$$

where $i = 0, 1, \cdots, L-1$ and $\mathbf{d}_k^i[(\mathbf{n})]$ and $\mathbf{c}_k^i[(\mathbf{n})]$ are respectively the k^{th} temporal high-pass and low-pass frames of the ith level decomposition. Note that the second line of Equation(2.2.5) accounts for the boundary values.

2.2.2 Motion-Compensated Frame Temporal Filtering (MC-FTF) Attack

We propose an extended FTF attack, called the *motion-compensated frame temporal filtering* (MC-FTF) attack using the lifting based MC-RTWT. The following are the motivating factors:

1) The low-pass temporal frames resulting from an MC-RTWT decomposition are of high visual quality.

2) The lifting based MC-RTWT can effectively exploit both the camera motion and the object motion by incorporating local motion estimation techniques.

3) The visual quality of the low-pass temporal frames can be improved by using adaptive update step in the lifting -based implementation of the MC-RTWT.

4) MC-RTWT gives the same number of attacked frames as the watermarked video.

In the proposed MC-FTF attack, the video frames are temporally filtered using the lifting based MC-RTWT. The low-pass temporal frames resulting from the MC-RTWT are expected to have good visual quality and these frames constitute the attacked video. The algorithm is performed in the following steps.

1. Segment the watermarked video frames into scenes. Let $\{\mathbf{y}_k[\mathbf{n}], \quad k = 0, 1, \cdots, N_f - 1\}$ represent the watermarked frames from a scene.

2. Apply a suitable motion-estimation technique to find the motion trajectories .

3. Select a suitable wavelet filter and decomposition level L. Apply the lifting-based MC-RTWT along the motion trajectories to get the low-pass frames $\{\mathbf{c}_k^L[\mathbf{n}], \quad k = 0, 1, \cdots, N_f - 1\}$.

The low-pass temporal frames resulting from the Lth decomposition level are thus obtained by low-pass filtering of the watermarked video frames along the motion trajectories and expected to have good visual quality. These frames constitute the attacked video and are given by

$$\hat{\mathbf{y}}_k[\mathbf{n}] = \mathbf{c}_k^L[\mathbf{n}], \qquad k = 0, 1, \cdots, N_f - 1 \ . \tag{2.2.7}$$

2.3 Performance Analysis

In this Section, we analyze the performance of the proposed attack in the case of the additive spread-spectrum watermarking scheme given in Equation (2.1.3). The following assumptions are made in the analysis.

A1) The motion model is of integer-pixel accuracy.

A2) The watermark embedding does not change the motion vectors. i.e., the motion vectors estimated from a pair of host frames are same as that estimated from the corresponding watermarked frames.

A3) The watermark frames $\{\mathbf{w}_k[\mathbf{n}]\}$ are statistically independent of the host frames.

If sub-pixel accurate motion model is used, then the prediction and the update lifting steps in Equations (2.2.5) and (2.2.6) involve interpolation steps. So, we assume a motion model with integer-pixel accuracy for mathematical tractability. Further, the Haar wavelet is used for simplifying the analysis.

Suppose the watermarked frames are decomposed up to one level of the MC-RTWT using the Haar filter. Then the high-pass temporal frames are given by

$$\begin{aligned}\mathbf{d}_k^1[\mathbf{n}] &= \mathbf{y}_k[\mathbf{n}] - \mathbf{y}_{k-1}[\mathcal{M}_{k-1\to k}(\mathbf{n})] \\ &= \mathbf{x}_k[\mathbf{n}] + \alpha\mathbf{w}_k[\mathbf{n}] - \Big(\mathbf{x}_{k-1}[\mathcal{M}_{k-1\to k}(\mathbf{n})] + \alpha\mathbf{w}_{k-1}[\mathcal{M}_{k-1\to k}(\mathbf{n})]\Big)\end{aligned}$$

and the low-pass temporal frames are given by

$$\begin{aligned}\mathbf{c}_k^1[\mathbf{n}] &= \mathbf{y}_k[\mathbf{n}] + \frac{1}{2}\mathbf{d}_{k+1}^1[\mathcal{M}_{k+1\to k}(\mathbf{n})] \\ &= \mathbf{x}_k[\mathbf{n}] + \alpha\mathbf{w}_k[\mathbf{n}] + \frac{\alpha}{2}\Big(\mathbf{w}_{k+1}[\mathcal{M}_{k+1\to k}(\mathbf{n})] - \mathbf{w}_k[\mathcal{M}_{k\to k+1}(\mathcal{M}_{k+1\to k}(\mathbf{n}))]\Big) \\ &\quad + \frac{1}{2}\Big(\mathbf{x}_{k+1}[\mathcal{M}_{k+1\to k}(\mathbf{n})] - \mathbf{x}_k[\mathcal{M}_{k\to k+1}(\mathcal{M}_{k+1\to k}(\mathbf{n}))]\Big) \ . \end{aligned} \tag{2.3.1}$$

If the motion model is invertible, i.e., $\mathcal{M}_{k\to m}\big(\mathcal{M}_{m\to k}(\mathbf{n})\big) = \mathbf{n}$,

$$\begin{aligned}\mathbf{c}_k^1[\mathbf{n}] &= \mathbf{x}_k[\mathbf{n}] + \frac{\alpha}{2}\Big(\mathbf{w}_k[\mathbf{n}] + \mathbf{w}_{k+1}[\mathcal{M}_{k+1\to k}(\mathbf{n})]\Big) + \frac{1}{2}\Big(\mathbf{x}_{k+1}[\mathcal{M}_{k+1\to k}(\mathbf{n})] - \mathbf{x}_k[\mathbf{n}]\Big) \\ &= \mathbf{x}_k[\mathbf{n}] + \frac{\alpha}{2}\Big(\mathbf{w}_k[\mathbf{n}] + \mathbf{w}_{k+1}[\mathcal{M}_{k+1\to k}(\mathbf{n})]\Big) - \frac{1}{2}\mathbf{e}_{k,k+1}[\mathbf{n}]\end{aligned}$$

where $\mathbf{e}_{k,m}[\mathbf{n}] = \mathbf{x}_k[\mathbf{n}] - \mathbf{x}_m[\mathcal{M}_{m\to k}(\mathbf{n})]$ is the motion-compensated prediction error. Under the assumption of the composition of motion operators, i.e., $\mathcal{M}_{k_1\to k_2}(\mathcal{M}_{k_2\to k_3}(\mathbf{n})) = \mathcal{M}_{k_1\to k_3}(\mathbf{n})$, it can be shown that the low-pass temporal frames resulting from L-level MC-RTWT decomposition are given by

$$\mathbf{c}_k^L[\mathbf{n}] = \mathbf{x}_k[\mathbf{n}] + \frac{\alpha}{2^L}\Big(\mathbf{w}_k[\mathbf{n}] + \sum_{i=1}^{2^L-1}\mathbf{w}_{k+i}[\mathcal{M}_{k+i\to k}(\mathbf{n})]\Big) + \frac{1}{2^L}\Big(\sum_{i=1}^{2^L-1}\mathbf{e}_{k,k+i}[\mathcal{M}_{k+i\to k}(\mathbf{n})]\Big). \tag{2.3.2}$$

If the motion model is able to capture the motion in the scene perfectly, then the energy of $\mathbf{e}_{k,m}[\mathbf{n}]$ is zero under perfect imaging conditions. In such a situation, increasing the decomposition level L increases the visual quality of the attacked video (except for the frames from a static scene carrying repetitive watermarks). But in practical situations, no motion model will be able to capture the motion perfectly. In general, as the temporal separation between the frames k and $k+i$ increases, the energy of the motion-compensated prediction error $\mathbf{e}_{k,k+i}[\mathbf{n}]$ also increases, there by reducing the visual quality of the attacked video.

Let the detection of the watermark be *non-blind* and the detection measure be the normalized correlation (NC) score between the extracted watermark from the attacked sequence and the original watermark. For the kth frame, the normalized correlation $\mathrm{NC_k}$ is defined as

$$\mathrm{NC_k} = \frac{1}{\alpha}\frac{\langle\hat{\mathbf{y}}_k[\mathbf{n}] - \mathbf{x}_k[\mathbf{n}], \mathbf{w}_k[\mathbf{n}]\rangle}{\|\mathbf{w}_k[\mathbf{n}]\|^2} \tag{2.3.3}$$

where $\hat{\mathbf{y}}_k[\mathbf{n}]$ is the attacked frame, $\langle\cdot\rangle$ is the *inner product* and

$$\|\mathbf{w}_k[\mathbf{n}]\|^2 = \langle\mathbf{w}_k[\mathbf{n}], \mathbf{w}_k[\mathbf{n}]\rangle \ . \tag{2.3.4}$$

The watermark detected form the attacked frame $\mathbf{c}_k^L$ in Equation (2.3.2) is given by

$$\mathrm{NC_k} = \frac{1}{\alpha}\frac{\langle\big(\mathbf{c}_k^L[\mathbf{n}] - \mathbf{x}_k[\mathbf{n}]\big), \mathbf{w}_k[\mathbf{n}]\rangle}{\|\mathbf{w}_k[\mathbf{n}]\|^2} \ . \tag{2.3.5}$$

Substituting the value of $\mathbf{c}_k^L$ from Equation (2.3.2) and using the additivity property of the inner product operator, we can obtain

$$\mathrm{NC_k} = \frac{1}{2^L} \frac{\langle \mathbf{w}_k[\mathbf{n}], \mathbf{w}_k[\mathbf{n}] \rangle + \sum_{i=1}^{2^L-1} \langle \mathbf{w}_{k+i}[\mathcal{M}_{k+i\to k}(\mathbf{n})], \mathbf{w}_k[\mathbf{n}] \rangle + \frac{1}{\alpha} \sum_{i=1}^{2^L-1} \langle \mathbf{e}_{k+i}[\mathbf{n}], \mathbf{w}_k[\mathbf{n}] \rangle}{\|\mathbf{w}_k[\mathbf{n}]\|^2}$$

$$\approx \frac{1}{2^L} + \frac{1}{2^L} \frac{\sum_{i=1}^{2^L-1} \langle \mathbf{w}_{k+i}[\mathcal{M}_{k+i\to k}(\mathbf{n})], \mathbf{w}_k[\mathbf{n}] \rangle}{\|\mathbf{w}_k[\mathbf{n}]\|^2} \quad . \tag{2.3.6}$$

In the above, $\langle \mathbf{e}_{k+i}[\mathbf{n}], \mathbf{w}_k[\mathbf{n}] \rangle \approx 0$ because the watermarks are statistically independent of the host frames.

In the following subsections, the dependency of attack performance on various factors like the watermarking scheme, motion-modelling and the estimated motion vectors are analyzed in detail.

2.3.1 Performance Against Different Watermarking Schemes

To analyze the dependence of the embedded watermarks on the attack performance, we consider the frame-by-frame watermark embedding and the TWT-domain watermarking scheme [SZT98].

Frame-by-frame additive spread-spectrum watermarking

In the frame-by-frame additive spread-spectrum watermarking given in Equation (2.1.3), there are mainly two embedding strategies : *independent* watermarks embedding and *repetitive* watermark embedding.

Independent watermarks embedding (SS watermarking): In this case, the embedded watermarks are statistically independent of each other. Therefore, $\mathbf{w}_{k+i}[\mathcal{M}_{k+i\to k}(\mathbf{n})]$ and $\mathbf{w}_k[\mathbf{n}]$ in Equation (2.3.6) will be uncorrelated no matter the amount of motion. Thus the attack will have similar effect as the FTF attack on the detectability of the watermark. However the visual quality of the attacked frames will be better than that of the FTF attack as the visually similar areas in successive frames are averaged.

Repetitive watermark embedding (SS-1 watermarking): In this case, the same watermark is embedded in all the frames, i.e., $\mathbf{w}_k = \mathbf{w}, \forall k$. Then Equation (2.3.6) can be rewritten as,

$$\mathrm{NC_k} \approx \frac{1}{2^L} + \frac{1}{2^L} \frac{\sum_{i=1}^{2^L-1} \langle \mathbf{w}[\mathcal{M}_{k+i\to k}(\mathbf{n})], \mathbf{w}[\mathbf{n}] \rangle}{\|\mathbf{w}[\mathbf{n}]\|^2} \quad . \tag{2.3.7}$$

If the frames are from a static scene, i.e., $\mathcal{M}_{k+i\to k}(\mathbf{n}) = \mathbf{n} \qquad \forall i, \mathbf{n}$, then

$$\mathrm{NC_k} \approx \frac{1}{2^L} + \frac{1}{2^L} \frac{\sum_{i=1}^{2^L-1} \langle \mathbf{w}[\mathbf{n}], \mathbf{w}[\mathbf{n}] \rangle}{\|\mathbf{w}[\mathbf{n}]\|^2} \approx 1 \ . \tag{2.3.8}$$

Thus the proposed attack is not effective in frames from static scenes embedded with repetitive watermarks. As the amount of motion in the frames increases, the correlation between $\mathbf{w}[\mathcal{M}_{k+i\to k}(\mathbf{n})]$ and $\mathbf{w}[\mathbf{n}]$ in Equation (2.3.7) decreases and hence the normalized correlation reduces. Since the spread-spectrum watermarks have very low spatial correlation, the second term in the RHS of Equation (2.3.7) depends only on the number of non-zero motion vectors and not on the absolute values of the motion vectors. So, the performance of the proposed attack against the SS-1 watermarking scheme depends on the moving areas in each frame.

TWT domain watermarking

In the TWT domain watermarking scheme, the watermark consists of *static* and *dynamic* temporal components.

$$\mathbf{y}_k[\mathbf{n}] = \mathbf{x}_k[\mathbf{n}] + \mathbf{w}_k[\mathbf{n}] = \mathbf{x}_k[\mathbf{n}] + \mathbf{w}^l[\mathbf{n}] + \mathbf{w}^h_k[\mathbf{n}] \tag{2.3.9}$$

where $\mathbf{w}^l[\mathbf{n}]$ and $\mathbf{w}^h_k[\mathbf{n}]$ are respectively the *static* and the *dynamic* temporal components of the watermark. Substituting $\mathbf{w}_k[n] = (\mathbf{w}^l[\mathbf{n}] + \mathbf{w}^h_k[\mathbf{n}])$ in Equation (2.3.6) and using the additivity property of the inner product, we can obtain

$$\mathrm{NC_k} \approx \frac{1}{2^L} + \frac{1}{2^L} \frac{\sum_{i=1}^{2^L-1} \langle \mathbf{w}^l[\mathcal{M}_{k+i\to k}(\mathbf{n})], \mathbf{w}^l[\mathbf{n}] \rangle + \langle \mathbf{w}^h_{k+i}[\mathcal{M}_{k+i\to k}(\mathbf{n})], \mathbf{w}^h_k[\mathbf{n}] \rangle}{\|\mathbf{w}^l[\mathbf{n}] + \mathbf{w}^h_k[\mathbf{n}]\|^2} \tag{2.3.10}$$

where the *static* and *dynamic* components of the watermark are statistically independent of each other, i.e., $\langle \mathbf{w}^l[\mathbf{n}], \mathbf{w}^h_i[\mathbf{n}] \rangle = 0, \forall i$. In a static scene, the watermark contains only the temporally static component ($\mathbf{w}^h_i[\mathbf{n}] = 0$) and the MC-FTF attack will not be effective. On the other hand, in a dynamic scene, the performance of the attack depends on the motion content of the scene. The attack performance on the static component of the watermark will be similar to that on the SS-1 watermark whereas the performance on the dynamic components will be similar to that on the SS watermark.

2.3.2 Impact of Motion Modelling

In the above analysis, we have assumed the *invertibility* and *composition* properties of the motion operators. But motion models like the block-based ones do not satisfy these properties. The impact of

these properties on the performance of the MC-TWT based video coding has been extensively studied [BKZV05]. The prediction and update steps in the lifting-based the MC-TWT require the estimation of *forward* and *backward* motion fields. Independent estimation of these motion fields from the corresponding frames will result in the *motion inversion error*. So, in all practical implementations of the MC-TWT based video coding, one motion field is estimated and the other is derived from it using *motion inversion techniques* [CW99, Kon05] to reduce the motion inversion error. Usually the motion field for the prediction step is estimated and that for the update step is obtained by inversion. Many motion inversion techniques like *collinear extension*, *neighbour-frame-copy*, and *nearest-neighbour motion inversion* have been proposed for the lifting-based MC-TWT [Kon05].

Consider the one level MC-RTWT decomposition using the Haar wavelet with sub-pixel accuracy motion. If the sub-pixel interpolation is linear, then Equation (2.3.1) can be rewritten as

$$\mathbf{c}_k^1[\mathbf{n}] = \mathbf{x}_k[\mathbf{n}] + \alpha\Big(\mathbf{w}_k[\mathbf{n}] - \frac{1}{2}\widetilde{\widetilde{\mathbf{w}}}_k[\mathcal{M}_{k\to k+1}(\mathcal{M}_{k+1\to k}(\mathbf{n}))]\Big) + \frac{\alpha}{2}\widetilde{\mathbf{w}}_{k+1}[\mathcal{M}_{k+1\to k}(\mathbf{n})] + \frac{1}{2}\Big(\widetilde{\mathbf{x}}_{k+1}[\mathcal{M}_{k+1\to k}(\mathbf{n})] - \widetilde{\widetilde{\mathbf{x}}}_k[\mathcal{M}_{k\to k+1}(\mathcal{M}_{k+1\to k}(\mathbf{n}))]\Big)$$

where the *tilde* notation is used to represent the spatially interpolated values at the sub-pixel points. Note that $\widetilde{\widetilde{\mathbf{w}}}_k[\mathbf{n}]$ represents the double interpolation (one during the prediction step and the other during update step). The corresponding NC score is given as

$$\mathrm{NC_k} \approx 1 - \frac{1}{2}\frac{\langle\widetilde{\widetilde{\mathbf{w}}}_k[\mathcal{M}_{k\to k+1}(\mathcal{M}_{k+1\to k}(\mathbf{n}))], \mathbf{w}_k[\mathbf{n}]\rangle + \langle\widetilde{\mathbf{w}}_{k+1}[\mathcal{M}_{k+1\to k}(\mathbf{n})], \mathbf{w}_k[\mathbf{n}]\rangle}{\|\mathbf{w}_k[\mathbf{n}]\|^2} \ .$$

Thus the detectability of the watermark depends on the correlation between $\widetilde{\widetilde{\mathbf{w}}}_k[\mathcal{M}_{k\to k+1}(\mathcal{M}_{k+1\to k}(\mathbf{n}))]$ and $\mathbf{w}_k[\mathbf{n}]$. The correlation between these terms depends on the number of pixels with motion inversion error, i.e., $\mathcal{M}_{k\to k+1}(\mathcal{M}_{k+1\to k}(\mathbf{n})) \neq \mathbf{n}$ and the interpolation method used. As the number of pixels with motion inversion error increases, the correlation between these terms decreases. As a result, the normalized correlation score increases.

2.3.3 Effect of Changes in Motion Vectors

In the MC-FTF attack, the attacker generally has to estimate the motion vectors from the watermarked sequence. However, the attacker may have access to the motion-vectors estimated from the corresponding host sequence. For example, if the watermarked sequence is in the compressed format, the attacker can obtain the coded motion vectors. We analyze the attack performance with respect to the motion vectors estimated from the host and the watermarked sequence. For the analysis, we consider the simple fixed size block-matching (FSBM) motion-estimation, where the *current frame* is partitioned into

non-overlapping macro-blocks of equal size and for each block, the best-matching block from the *reference* frame is calculated. In order to find the best-matching block, a distortion measure such as the *mean absolute difference* (MAD) or the *mean-square error* (MSE) is generally used. For simplifying the analysis, the MSE is used here as the distortion measure.

Let $\mathbf{x}_{k_1}$ and $\mathbf{x}_{k_2}$ be respectively, the reference frame and current frame from a host the sequence. Consider a macro-block in the current frame with the set $\mathcal{B}$ representing the corresponding pixel locations. The best-matching block in the reference frame is the one which minimizes the distortion function:

$$D^{\mathbf{x}}(\mathbf{m}) = \frac{1}{|\mathcal{B}|}\sum_{\mathbf{n}\in\mathcal{B}}\big(\mathbf{x}_{k_2}[\mathbf{n}] - \mathbf{x}_{k_1}[\mathbf{n}+\mathbf{m}]\big)^2 \tag{2.3.11}$$

where $\mathbf{m} \in \{(i,j), \quad -P \le i,j \le P\}$, P is the maximum displacement and $|\mathcal{B}|$ is the cardinality of $\mathcal{B}$. The motion vector $\mathbf{m}_0$ for the block $\mathcal{B}$ is given as

$$\mathbf{m}_0 = \arg\min_{\mathbf{m}} D^{\mathbf{x}}(\mathbf{m}) \ . \tag{2.3.12}$$

Consider the estimation of the motion vector for the same block after embedding an additive watermark given in Equation (2.1.1). The estimated motion vector $\mathbf{m}_0'$ corresponding to the same block in the reference frame is now given as

$$\mathbf{m}_0' = \arg\min_{\mathbf{m}} D^{\mathbf{y}}(\mathbf{m}) \tag{2.3.13}$$

where

$$\begin{aligned} D^{\mathbf{y}}(\mathbf{m}) &= \frac{1}{|\mathcal{B}|}\sum_{\mathbf{n}\in\mathcal{B}}\big(\mathbf{y}_{k_2}[\mathbf{n}] - \mathbf{y}_{k_1}[\mathbf{n}+\mathbf{m}]\big)^2 \\ &= \frac{1}{|\mathcal{B}|}\sum_{\mathbf{n}\in\mathcal{B}}\Big(\big(\mathbf{x}_{k_2}[\mathbf{n}] - \mathbf{x}_{k_1}[\mathbf{n}+\mathbf{m}]\big) + \alpha\big(\mathbf{w}_{k_2}[\mathbf{n}] - \mathbf{w}_{k_1}[\mathbf{n}+\mathbf{m}]\big)\Big)^2 \ . \end{aligned} \tag{2.3.14}$$

Since the watermark frames are independent of the host frames, the cross-terms in the expansion of the RHS of Equation (2.3.14) can be neglected and can simplified as

$$D^{\mathbf{y}}(\mathbf{m}) \simeq D^{\mathbf{x}}(\mathbf{m}) + D^{\mathbf{w}}(\mathbf{m}) \tag{2.3.15}$$

where

$$D^{\mathbf{w}}(\mathbf{m}) = \frac{\alpha^2}{|\mathcal{B}|}\sum_{\mathbf{n}\in\mathcal{B}}\big(\mathbf{w}_{k_2}[\mathbf{n}] - \mathbf{w}_{k_1}[\mathbf{n}+\mathbf{m}]\big)^2 \ .$$

Let us now consider the relation between $\mathbf{m}_0$ and $\mathbf{m}_0'$ for different watermarking schemes. In the

case of SS-1 watermarking scheme ($\mathbf{w}_{k_1} = \mathbf{w}_{k_2}$), it can be shown that

$$D^{\mathbf{y}}(\mathbf{m}) = \begin{cases} D^{\mathbf{x}}(\mathbf{m}) + 2\alpha^2, & \mathbf{m} \neq \mathbf{0} \\ D^{\mathbf{x}}(\mathbf{m}), & \mathbf{m} = \mathbf{0} \end{cases}$$

and

$$\min D^{\mathbf{y}}(\mathbf{m}) = \min \left(D_{\mathbf{x}}(\mathbf{0}), D_{\mathbf{x}}(\mathbf{m}_0) + 2\alpha^2 \right) .$$

The motion vector $\mathbf{m}_0'$ is now given as

$$\mathbf{m}_0' = \begin{cases} \mathbf{0}, & \left(D_{\mathbf{x}}(\mathbf{0}) - D_{\mathbf{x}}(\mathbf{m}_0) \right) < 2\alpha^2 \\ \mathbf{m}_0, & \text{otherwise} \end{cases} .$$

Thus, a non-zero motion vector may become *zero* when estimation is done from the watermarked sequence. This *bias* towards zero motion vector will depend on the embedding strength of the watermark and the content of the block. It will be more prominent in the blocks in *smooth* areas where the difference between $D_{\mathbf{x}}(\mathbf{0})$ and $D_{\mathbf{x}}(\mathbf{m}_0)$ is generally small. Thus, if the motion is estimated from the watermarked sequence, the number of zero motion vectors will be always greater than or equal to that estimated from the host. As we have shown earlier, the NC performance of the attack against the SS-1 watermarking improves (NC value decreases) as the number of non-zero motion vectors increases. So, the NC performance of the attack will be better with the motion-vectors estimated from the host sequence as compared to that with motion-vectors estimated from the watermarked sequence. This fact has been verified experimentally in the next Section.

In the case of the SS watermarking, we can obtain

$$D^{\mathbf{x}}(\mathbf{m}) = D_{\mathbf{x}}(\mathbf{m}) + 2\alpha^2, \quad \forall \mathbf{m}$$

and

$$\min D^{\mathbf{y}}(\mathbf{m}) = D^{\mathbf{x}}(\mathbf{m}_0)$$

Thus, there is no change in the motion vectors estimated before and after the addition of SS watermark ($\mathbf{m}_0' = \mathbf{m}_0$).

2.4 Experimental Results

In order to evaluate the performance of the proposed attack, experiments are conducted on a number of test video sequences. The results for the *Antibes*, *Foreman*, *Coastguard*, *Mobile* and the *Stefan* sequences are reported. The *Antibes* is a synthetic sequence created from a panoramic image with

horizontal translation of 2 pixels per frame from right to left. The other sequences are standard MPEG test sequences. Each frame of the *Antibes* and the *Stefan* sequences is of size 352×240 and the other sequences consist of frames of size 352×288. The first 64 frames from each sequence are used in the experiments.

The performance of the attack is tested with the sequences marked using two cases of the frame-by-frame watermarking scheme given in Subsection 2.3.1 : SS watermarking and SS-1 watermarking. The embedding strength α of the additive watermark is chosen such that the *peak signal-to-noise ratio* (PSNR), defined as

$$\text{PSNR} = 10\log_{10}\left(\frac{1}{N_f \cdot N_1 \cdot N_2}\sum_{k=1}^{N_f}\sum_{n_1=1}^{N_1}\sum_{n_2=1}^{N_2}\frac{255^2}{(\mathbf{y}_k[n_1,n_2]-\mathbf{x}_k[n_1,n_2])^2}\right), \tag{2.4.1}$$

is around 38 dB. The normalized correlation between the extracted watermark from the attacked sequence and the original watermark defined in Equation (2.3.3) is used as the detection score. The visual quality of the attacked video is evaluated in terms of the PSNR performance of the attacked sequence with respect to the host.

The watermarked video frames are subjected to the MC-FTF attack using the Haar filter. Two different motion-estimation algorithms are employed to obtain the motion vectors. One technique uses the simple *fixed size block matching*(FSBM) with a block-size of 16×16 and the MSE as the distortion metric. The other technique uses the *hierarchical variable size block matching* (HVSBM) technique, with the macro-block size of 64×64 and the block sizes down to 4×4 [CW99]. Motion compensation with *integer-pixel*, *half-pixel* and *quarter-pixel* accuracies are considered in the experiments. The intensity values at the sub-pixel locations are computed with the *cubic-spline interpolation*. Motion vectors are estimated for the prediction step and for the update step, they are obtained by the *nearest-neighbor inversion* of these estimated motion vectors [Kon05]. The MC-FTF attack is tested using the MC-RTWT with and without the adaptive update step. All the experiments were performed with 20 randomly generated watermarks and the average values are presented. The following Subsections detail the experimental results.

2.4.1 Performance Against SS Watermarking

In the first set of experiments, the performance of the attack against the SS watermarks is evaluated. Table 2.1 shows the performance of the proposed attack for different levels of the MC-RTWT decomposition with integer-pixel motion estimation. The reported NC values are the average over all the

frames in the corresponding sequences. As the number of MC-RTWT decomposition levels increases, the watermark detectability in terms of the NC value reduces. Except for the *Antibes* sequence, the visual quality of the attacked sequence decreases with the increasing level of decomposition. These results are in accordance with the analysis presented in the previous section. Further, due to better motion-estimation of the HVSBM as compared to the FSBM, in all levels of MC-RTWT decomposition, the visual quality of the attacked sequence is better with the HVSBM as compared to that with the FSBM. The level of decomposition of the MC-RTWT is a compromise between the visual quality of the attacked sequence and the watermark detectability. In all the subsequent experiments reported in this Section, 3-level decomposition for the MC-RTWT is considered. The high PSNR values and low NC scores demonstrate the effectiveness of the MC-FTF attack.

Table 2.2 details the performance of the MC-FTF attack with varying accuracies of the motion-compensation. It is observed that increasing the accuracy of motion vectors increases the visual quality of the attacked sequence. However, the attack performance slightly decreases (high NC values) with the increase in the motion accuracy. This is due to the interpolation in sub-pixel accurate motion-compensation and motion inversion error in the update lifting step of the MC-RTWT. Also note that the NC values for the *Antibes* are not affected by the change in precision because the sequence contains only integer-pixel motion. It is observed that the changes in the PSNR and the NC values are significant between the integer-pixel and half-pixel accurate motion-compensations, whereas the values are marginally affected by changing the accuracy from half-pixel to quarter-pixel. The changes in PSNR and NC values are similar for the two types of motion-estimation.

Finally, the performance of the attack with the adaptive update step of the MC-RTWT is presented in Table 2.3. For all precisions of motion-compensation accuracy, the adaptive update step significantly improves the visual quality of the attacked sequence. This is evident from a comparison of the PSNR values given in Table 2.2 and Table 2.3. At the same time, the NC performance slightly decreases with the adaptive update step. Figures 2.1 and 2.2 show one watermarked frame and the corresponding attacked frame, from the *Stefan* and the *Foreman* sequence respectively. The good visual quality of the attacked video with the adaptive update step is evident from these figures.

2.4.2 Performance Against SS-1 Watermarking

The first experiment on sequences carrying the SS-1 watermark studies the change in the attack performance with motion-vectors estimated from the host sequences and those from the watermarked

sequences. The PSNR and NC values of the attacked sequences with 3-levels of MC-RTWT decomposition and varying precision of motion estimation are presented in Tables 2.4 and 2.5. The NC scores of the attacked sequences with motion-vectors estimated form the watermarked sequences are higher than those with motion estimation from the host sequences. The change is significant for the sequences having smooth areas(*Antibes* and *Foreman*) whereas it is negligible for the *Mobile* sequence which contains more spatial details. This change in the NC values validates the analysis presented in the previous Section. Note that the PSNR values corresponding to the motion-vectors estimated from the watermarked sequences are lower than those corresponding to the motion-vectors estimated from the host sequences. As we have seen, more watermark components are removed in the first case compared to the second. This explains the change in the PSNR values of the attacked sequences which is computed with reference to the host sequences. The attack performance with the adaptive update step is shown in Table 2.6. As expected, the adaptive update step significantly increases the PSNR performance of the attacked video with a slight decrease in the NC performance. The PSNR performance of the attacked sequence is better in the case of the HVSBM motion-estimation as compared to the FSBM.

On comparing the performance of the attack against the SS and SS-1 watermarking schemes, it can be observed that the attack is more effective against the SS watermark. As shown in our earlier analysis, the performance of the attack against the SS-1 watermarking depends on the number of non-zero motion-vectors whereas it is independent of motion in the case of the SS watermarking. Comparative plots of the NC scores of each attacked frame from three sequences are depicted in Figure 2.3. It is observed that the shapes of the NC plots are almost constant in the case of SS watermarking, whereas they vary with sequences in SS-1 watermarking. Due to the uniform motion in the *Mobile* sequence, the NC values are almost constant. However, a considerable variation of the frame-wise NC values is observed in the case of the *Foreman* and the *Stefan* sequences. It is due to the nonuniform motion in different frames of these sequences. In particular, near-unity peaks around frames 30 and 54 of the *Foreman* sequence and frame 30 of the *Stefan* sequence are because of the negligible camera motion. In these cases, the NC performance depends only on the area of the moving objects. The *Foreman* sequence with the foreground area larger than that of the *Stefan* sequence shows better NC performance near these peaks. Also, note the difference in the NC performance against SS-1 watermarking for attacks with the motion-vectors estimated from the watermarked sequences and those from the host sequences.

Sequence	Level 1		Level 2		Level 3	
	PSNR(dB)	NC	PSNR(dB)	NC	PSNR(dB)	NC
Antibes	39.38	0.50	39.30	0.26	39.39	0.14
Foreman	37.69	0.53	35.54	0.29	33.63	0.17
Coastguard	35.52	0.51	32.10	0.27	30.10	0.16
Mobile	30.88	0.51	28.65	0.26	27.03	0.14
Stefan	31.54	0.53	28.87	0.29	26.74	0.18

(a)

Sequence	Level 1		Level 2		Level 3	
	PSNR(dB)	NC	PSNR(dB)	NC	PSNR(dB)	NC
Antibes	40.50	0.51	41.51	0.26	42.45	0.14
Foreman	37.88	0.53	36.25	0.30	34.48	0.18
Coastguard	36.51	0.51	33.07	0.27	30.83	0.16
Mobile	31.54	0.51	29.98	0.26	28.34	0.14
Stefan	32.66	0.53	30.29	0.3	28.67	0.18

(b)

Table 2.1: Performance of the MC-FTF attack on SS watermarked sequences for different levels of the MC-RTWT decomposition. The motion-vectors are estimated using (a) the FSBM and (b) the HVSBM.

Sequence	integer-pixel		half-pixel		quarter-pixel	
	PSNR(dB)	NC	PSNR(dB)	NC	PSNR(dB)	NC
Antibes	39.39	0.14	41.4	0.14	41.63	0.14
Foreman	33.63	0.17	33.81	0.3	33.83	0.3
Coastguard	30.1	0.16	30.62	0.24	30.87	0.26
Mobile	27.03	0.14	28.32	0.24	28.75	0.24
Stefan	26.74	0.18	27.76	0.28	27.91	0.29

(a)

Sequence	integer-pixel		half-pixel		quarter-pixel	
	PSNR(dB)	NC	PSNR(dB)	NC	PSNR(dB)	NC
Antibes	42.45	0.14	43.1	0.14	43.22	0.14
Foreman	34.48	0.18	35.02	0.3	35.09	0.3
Coastguard	30.83	0.16	31.79	0.25	32.17	0.26
Mobile	28.34	0.14	30.29	0.25	30.95	0.24
Stefan	28.67	0.18	30.61	0.27	31.02	0.28

(b)

Table 2.2: Performance of the MC-FTF attack on SS watermarked sequences for varying precision of motion-estimation. The motion-vectors are estimated using (a) the FSBM and (b) the HVSBM.

Sequence	integer-pixel		half-pixel		quarter-pixel	
	PSNR(dB)	NC	PSNR(dB)	NC	PSNR(dB)	NC
Antibes	45.27	0.15	46.09	0.15	46.17	0.15
Foreman	37.28	0.21	37.33	0.33	37.45	0.33
Coastguard	33.06	0.21	33.5	0.28	33.71	0.29
Mobile	30.94	0.19	32.2	0.27	32.79	0.27
Stefan	30.82	0.23	32.09	0.31	32.39	0.32

(a)

Sequence	integer-pixel		half-pixel		quarter-pixel	
	PSNR(dB)	NC	PSNR(dB)	NC	PSNR(dB)	NC
Antibes	46.39	0.16	46.45	0.17	46.44	0.17
Foreman	37.84	0.22	38.17	0.33	38.33	0.33
Coastguard	33.69	0.21	34.52	0.29	34.88	0.3
Mobile	32.14	0.2	33.76	0.28	34.45	0.26
Stefan	32.48	0.24	34.24	0.31	34.72	0.31

(b)

Table 2.3: Performance of the MC-FTF attack with adaptive update step on SS-watermarked sequences for varying precision of motion-estimation. The motion vectors are estimated using (a) the FSBM and (b) the HVSBM.

Sequence	integer-pixel		half-pixel		quarter-pixel	
	PSNR(dB)	NC	PSNR(dB)	NC	PSNR(dB)	NC
Antibes	39.34	0.15	41.33	0.15	41.56	0.15
Foreman	33.51	0.24	33.67	0.37	33.68	0.38
Coastguard	30.06	0.23	30.56	0.33	30.79	0.34
Mobile	27	0.18	28.29	0.29	28.72	0.27
Stefan	26.7	0.31	27.71	0.42	27.85	0.44

(a)

Sequence	integer-pixel		half-pixel		quarter-pixel	
	PSNR(dB)	NC	PSNR(dB)	NC	PSNR(dB)	NC
Antibes	37.71	0.39	38.85	0.42	38.96	0.42
Foreman	33.22	0.38	33.31	0.54	33.29	0.55
Coastguard	29.98	0.29	30.46	0.41	30.67	0.43
Mobile	26.97	0.19	28.27	0.31	28.71	0.3
Stefan	26.66	0.42	27.67	0.52	27.8	0.53

(b)

Table 2.4: Performance of the MC-FTF attack on SS-1 watermarked sequences for varying precision of motion-estimation. The motion-vectors are estimated using the FSBM from (a) the host sequence and (b) the watermarked sequence.

Sequence	integer-pixel		half-pixel		quarter-pixel	
	PSNR(dB)	NC	PSNR(dB)	NC	PSNR(dB)	NC
Antibes	42.4	0.14	43.04	0.15	43.17	0.14
Foreman	34.34	0.25	34.84	0.37	34.91	0.38
Coastguard	30.79	0.21	31.73	0.3	32.1	0.31
Mobile	28.31	0.17	30.25	0.28	30.91	0.26
Stefan	28.61	0.3	30.51	0.4	30.9	0.41

(a)

Sequence	integer-pixel		half-pixel		quarter-pixel	
	PSNR(dB)	NC	PSNR(dB)	NC	PSNR(dB)	NC
Antibes	40.94	0.25	40.85	0.32	40.95	0.32
Foreman	33.81	0.41	34.26	0.56	34.29	0.58
Coastguard	30.73	0.25	31.66	0.37	32.01	0.39
Mobile	28.18	0.21	30.21	0.32	30.87	0.3
Stefan	28.49	0.4	30.37	0.51	30.75	0.52

(b)

Table 2.5: Performance of the MC-FTF attack on SS-1 watermarked sequences for varying precision of motion-estimation. The motion-vectors are estimated using the HVSBM from (a) the host sequence and (b) the watermarked sequence.

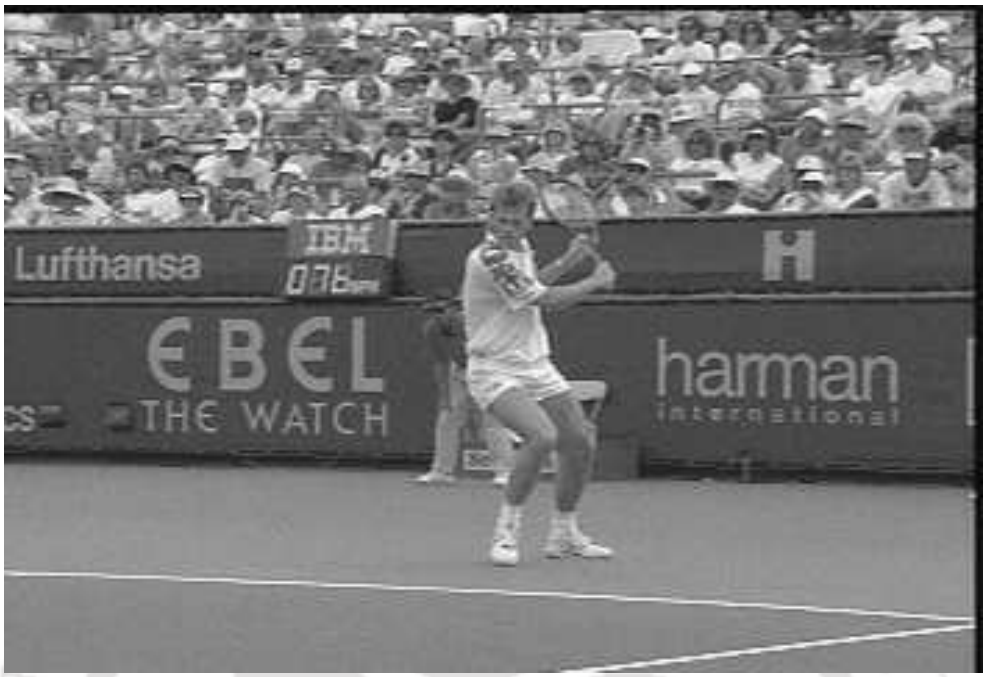
Sequence	integer-pixel		half-pixel		quarter-pixel	
	PSNR(dB)	NC	PSNR(dB)	NC	PSNR(dB)	NC
Antibes	40.86	0.4	40.98	0.42	40.97	0.43
Foreman	36.41	0.41	36.36	0.56	36.36	0.58
Coastguard	32.83	0.33	33.2	0.44	33.37	0.46
Mobile	30.82	0.24	32.08	0.34	32.67	0.32
Stefan	30.63	0.46	31.85	0.54	32.11	0.55

(a)

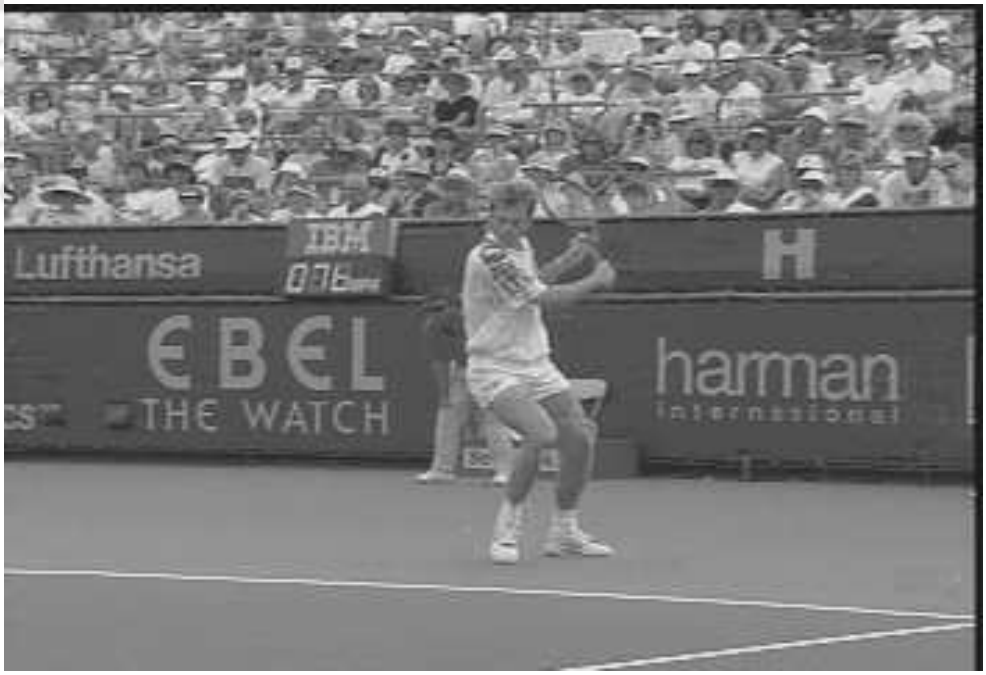
Sequence	integer-pixel		half-pixel		quarter-pixel	
	PSNR(dB)	NC	PSNR(dB)	NC	PSNR(dB)	NC
Antibes	43.56	0.28	42.76	0.34	42.78	0.35
Foreman	36.65	0.44	36.82	0.59	36.85	0.6
Coastguard	33.53	0.3	34.3	0.42	34.6	0.43
Mobile	31.89	0.26	33.56	0.36	34.25	0.32
Stefan	32.14	0.46	33.77	0.55	34.15	0.55

(b)

Table 2.6: Performance of the MC-FTF attack with the adaptive update step on SS-1 watermarked sequences. The motion-vectors are estimated from the watermarked sequences using: (a) the FSBM and (b) the HVSBM.



(a)



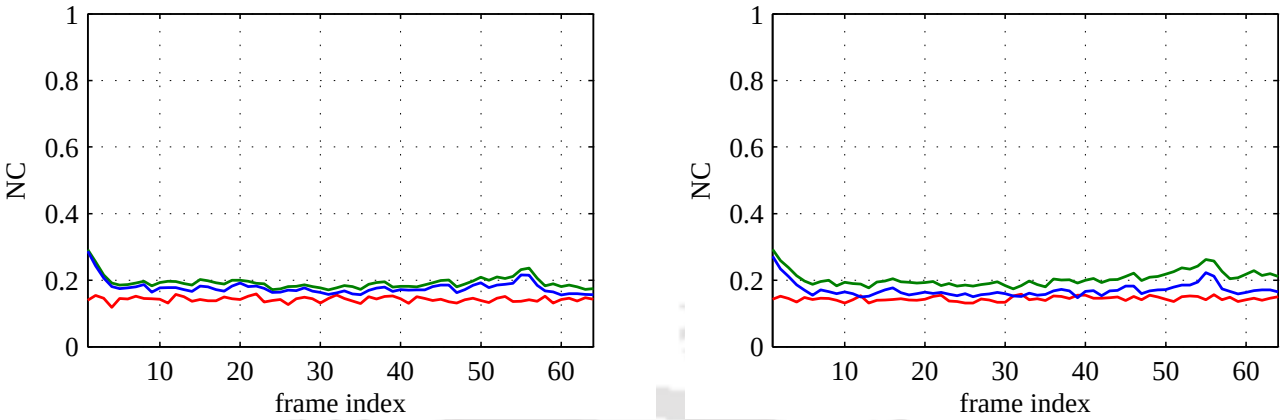
(b)

Figure 2.1: A sample frame from the *Stefan* sequence: (a) the watermarked frame and (b) the attacked frame.

(a)

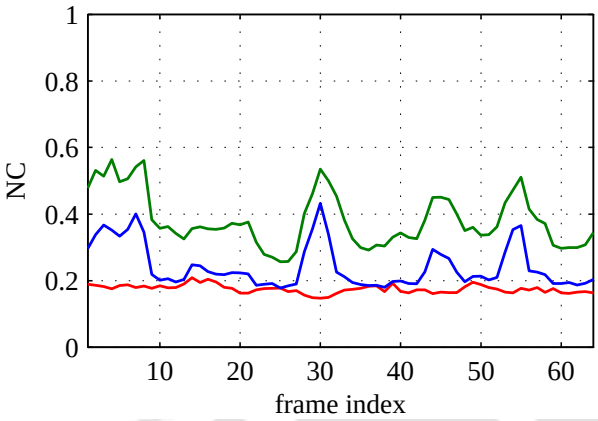
(b)

Figure 2.2: A Sample frames from the *Foreman* sequence: (a) the watermarked frame and (b) the attacked frame.

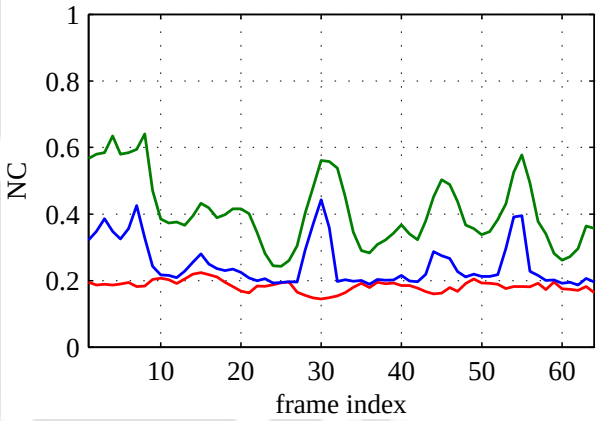


(a) Mobile (FSBM)

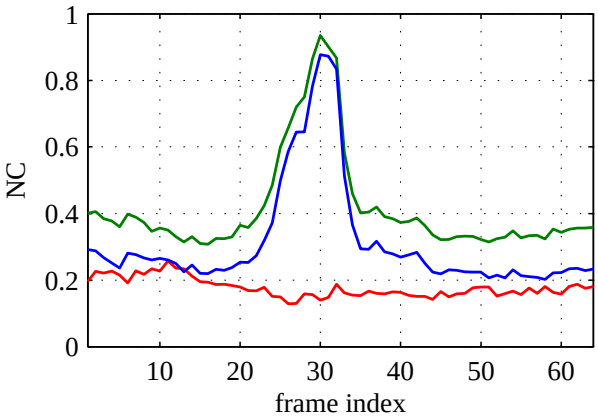
(b) Mobile (HVSBM)



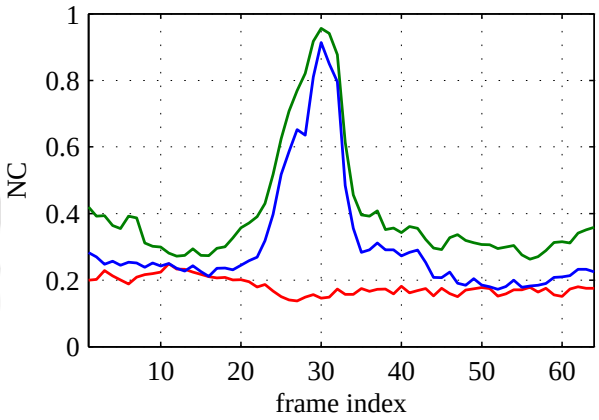
(c) Foreman (FSBM)



(d) Foreman (HVSBM)



(e) Stefan (FSBM)



(f) Stefan (HVSBM)

Figure 2.3: Detector performance of the SS and SS-1 watermarking schemes after subjected to MC-FTF attack. The red lines correspond to SS watermarking. The green (resp. blue) line corresponds to the NC performance of the attack using estimated motion-vectors from the watermarked (resp. host) sequences against SS-1 watermarking.

2.5 Discussion

The inter-frame collusion is a powerful attack against most of the existing watermarking schemes, particularly against the frame-by-frame watermark embedding. The existing countermeasures against the inter-frame collusion attacks overlook the motion in the video sequence. As a result, attackers can exploit the motion information to defeat such schemes. One such attack is to align neighboring frames in a video sequence according to estimated motion vectors and then apply temporal filtering to remove uncorrelated watermarks along the motion-trajectories.

This Chapter proposed a new motion-compensated inter-frame collusion attack. In the proposed MC-FTF attack, the watermarked frames are subjected to the MC-RTWT decomposition and the resulting low-pass temporal frames constitute the attacked video. A content-adaptive update lifting step significantly improves the visual quality of the video in those areas where the motion-estimation fails. The experimental results confirm the effectiveness of the attack in removing the watermark with a good visual quality of the attacked video. We have analyzed the dependency of the attack performance on various factors like the motion-model and accuracy of motion estimation. It has been shown that the embedded watermark biases the motion-estimation and if the attacker uses the motion-vectors obtained from the host sequence, the attack performance will be better. Even though we have used only the simple block-based motion estimation in the experiments, by using more advanced motion-models like the mesh-based model, it could be possible to obtain better quality attacked video. Note that in the emerging MC-TWT based scalable video coding, the reduced frame rate video consists of low-pass temporal wavelet frames. So, if a watermarked video sequence is coded with such a coding technique, the embedded watermark will undergo the *unintentional* MC-FTF attack.

Chapter 3

Motion-Coherent Watermarking of Compressed Video

The previous chapter evaluated the performance of the MC-FTF attack. The MC-FTF attack succeeds in removing uncorrelated watermarks along the motion-trajectories. As pointed out earlier, its success lies in exploiting the motion in the video frames which is not considered during watermark embedding. The motion-compensation step in the MC-FTF attack aligns *similar* regions in different frames before temporal filtering. If these *similar* regions contain uncorrelated watermarks, the subsequent temporal filtering attenuates them. So, any countermeasure against the MC-FTF attack should incorporate the motion information during watermark embedding so that the similar areas along the motion trajectories carry correlated watermarks. Such a watermark is *coherent* with the motion and the corresponding watermarking scheme is known as the *motion-coherent* (MC) watermarking [Doë05]. This chapter addresses the problem of developing computationally efficient MC watermarking schemes for compressed video. The chapter first reviews earlier work on watermarking using motion information, including the MC watermarking. A brief description of video compression techniques and a review of compressed-domain watermarking schemes are then presented. Finally, the proposed MC watermarking schemes are presented along with simulation results.

3.1 Watermark Embedding Using Motion Information

Motion information has long been exploited to develop efficient video compression techniques. However, only a few video watermarking algorithms have utilized the motion information. The initial attempts to incorporate motion information during watermark embedding were aimed at improving

the visual quality of the watermarked sequence. Video watermarking introduces some visual artifacts which are not visible in a single frame of the video, but start appearing when the video is played in the continuous time. For example, embedding the same watermark pattern in successive frames of video introduces an artifact, known as the *dirty window effect* [MH00]. The static watermark pattern in successive frames causes an impression that the objects are moving behind a dirty window. Embedding uncorrelated watermarks in successive frames is not a solution to this problem since such an approach introduces the visually annoying *flicker artifact*. There are two different approaches proposed in literature to reduce the visual artifacts in the watermarked sequence. In the first approach, the watermarks embedded in different frames are no longer static, but move with the motion in the frames. In the second approach, the watermarks remains static while the embedding strengths are locally modified according to motion.

In [LOL00], Lee *et al.* proposed a solution which belongs to the first approach. In this scheme, the first frame in each scene is divided into non-overlapping blocks of size 16×16 pixels. For each block in the first frame, the best and the second best matching blocks in the next frame are computed using the *displaced frame difference* (DFD). A block is chosen for watermark embedding only if the corresponding motion vector is non-zero and the difference between the DFDs corresponding to the best matching and the second best matching blocks are greater than a predefined threshold. Thus the selection of blocks for embedding is based on their motion and the texture/edge contents. The selected blocks are transformed to 2-D wavelet domain and the watermark is embedded in the mid-frequency bands. The watermarked blocks are tracked using the motion estimation and the same watermark is embedded in the matching blocks. This process continues till the end of the scene.

In the watermarking scheme proposed by the researchers from the British Broadcasting Corporation (BBC) [WMS04], the watermark embedded in the consecutive frames are moved to follow the average motion in the frames. The average motion between two frames is estimated by calculating the cross-correlation between the frames. The watermark embedded in a frame is the spatially shifted (according to motion) version of that embedded in the previous frame. An error frame is then calculated by taking the difference between the current and the shifted version of the previous frame. Finally, the shifted watermark is modulated according to the error frame and added to the current frame.

A *motion sensitive* watermarking scheme has been proposed by Schimmel [Sch01]. In this scheme, the same watermark pattern is embedded in different frames. However, the embedding strength of watermark in each frame is locally scaled according to motion. They observed that the extent of perceived dirty window effect depends on the velocity of moving areas. The human visual system perceives the

effect more significantly in the low-velocity areas as compared to those with high velocity. The watermark embedding algorithm takes this factor into consideration in such a way that the embedding strength is a locally varying function with lower values in low velocity areas and higher values in static and high velocity areas.

The design goal of these watermarking schemes, developed prior to the introduction of the MC-FTF attack, was to improve the visual quality of the watermarked sequence. Nevertheless, it is worthwhile to assess these methods in the light of MC-FTF attack. In [LOL00], the watermark is embedded along the motion trajectory and the scheme will survive the MC-FTF attack. The drawback of this scheme is that only a few moving blocks which have significant texture/edge content are chosen for watermark embedding, thereby greatly reducing the embedding capacity. The other two schemes do not embed correlated watermarks along the motion-trajectory and will fail against the MC-FTF attack. In the method proposed by the BBC [WMS04], the watermark is coherent with *average* translational motion between the frames. So the watermark in those areas whose motion coincides with the average motion will survive the MC-FTF attack. This method, in general, does not guarantee the resistance to the MC-FTF attack. The watermark generated by Schimmel's technique, being temporally static, performs similar to the SS-1 watermark described in the previous chapter, i.e., survives the attack only in static scenes.

3.1.1 Motion-Coherent Watermarking

In the pioneering work, Doërr *et al.* [DD04a] investigated how the motion information can be effectively exploited to counter the MC-FTF attack. The basic idea behind the watermarking scheme is to compensate for camera motion before the watermark embedding to generate an MC watermark. The video frames are the 2-D projections of the 3-D scene at different time instances. Each video scene generally consists of a background, which is constant throughout the entire duration of the scene and a number of moving objects. The projected background in each frame depends on the motion of the camera. If the camera is static, the same portion of the 3-D background is projected into all the frames, except for the areas covered by moving objects. On the other hand, if the camera is moving, different portions of the background are projected into different frames. The first step in the watermarking process is to generate a *mosaic* representation of the background, which provides a snapshot view of the scene. The moving objects in the video frames are separated from the background and the resulting background frames are then registered with a reference frame to generate the mosaic. A first-order poly-

nomial motion-model is assumed for the motion in the scene and the model parameters are estimated from the frames. The quality of the generated mosaic depends on the ability of the assumed motion model to capture the *true motion*. Once the mosaic representation is obtained, a key dependent watermark, having the same size of the mosaic is added to the mosaic. The resulting mosaic is registered back using the same parameters used for the mosaic generation and the moving objects are inserted back to get the watermarked sequence. However the double interpolation due to the registration and inverse registration steps affects the detector performance. To solve this problem, the authors proposed an alternative method [Doë05], the details of which is shown in Figure 3.1[1]. In this method, the portion of the watermark that corresponds to the background of each frame is registered back and added to the background to get the watermarked frames. The scheme ensures that the same areas in the background carries the same watermark whenever it appears in the video, thereby generating a motion-coherent watermark for the background.

Even though the scheme embeds *ideal* motion-coherent watermark in the background region, it has the following limitations:

- The estimation of the frame registration parameters involves high computational cost, which prevents the use of the watermarking scheme in real-time applications
- It is not applicable to generic video since good quality mosaics can be generated only when there is no or little local motion [IAH95, TRS96]
- The watermark is added only to the background portion, thereby reducing the embedding capacity and leaving the moving objects unprotected.

Harmanci and Mihacak [HM05] proposed a watermarking scheme based on the linear statistics of overlapping random regions in each frame. The weights of the linear statistics are pseudo-randomly computed for the initial frame. For the subsequent frames, these weights are updated by a time-averaging AR(1) process along the motion trajectories estimated using an *optical flow* algorithm. The watermark is obtained by solving a constrained optimization problem such that the linear statistics of the marked frame is equal to the key-controlled quantized version of the of the corresponding host frame. A regularization procedure ensures the smoothness of the watermarked sequence along the motion trajectories. Unlike in the mosaic-based watermarking scheme, assessing the motion-coherency in

[1] Reproduced from [Doë05] with the permission of the author.

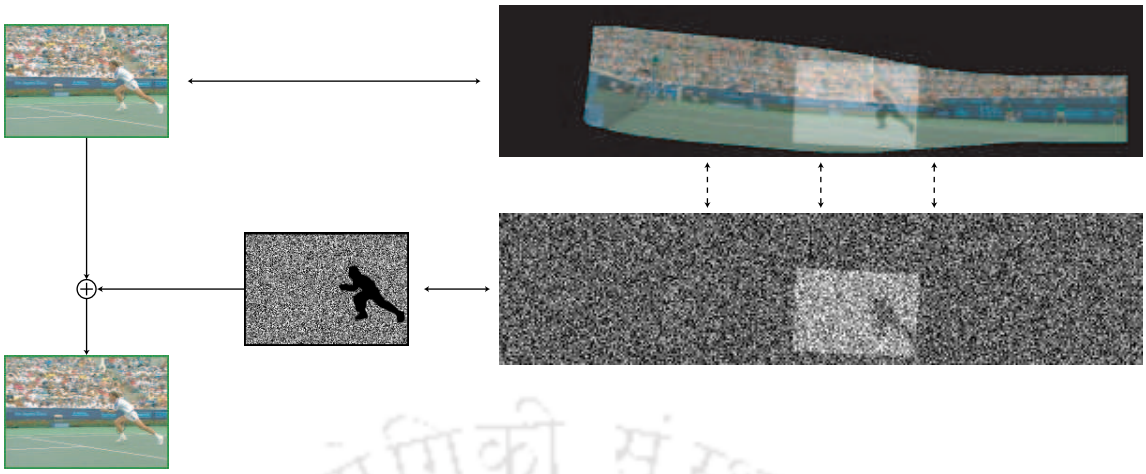


Figure 3.1: Mosaicing-based watermarking

the generated watermark is not straight-forward. However, their reported experimental results indicate the motion-coherency in the watermark.

The motion-coherent watermarking schemes discussed above operate in the uncompressed domain. However in all the practical applications, video sequences are stored and transmitted in the compressed format to save the storage space and the transmission bandwidth. In such situations, any watermarking scheme operating in the uncompressed domain needs to fully decompress the sequence, embed watermark and finally recompress the watermarked sequence. Due to the high computational cost, the decompress-watermarking-recompress strategy is not suitable for many applications like the fingerprinting in video-on-demand [LPKD01], where different watermarks need to be added in different copies of the video. Moreover, the motion-coherent watermarking schemes have additional computational burden of estimating the motion parameters.

An alternative and computationally efficient approach to the MC watermarking may be to exploit the motion information available in a compressed sequence. For example, generating mosaics by using the motion information available in the MPEG stream has been actively investigated [Pil97, JDD99, BR02]. These techniques use the available motion information in the MPEG stream to estimate the camera motion parameters, thereby significantly reducing the computational cost. The mosaic-based watermarking scheme can be modified by incorporating these techniques. Note that even with such techniques, the mosaic-based scheme needs complete decompression of the sequence before watermark embedding to generate the mosaic.

The drawbacks of the existing implementations of the MC watermarking schemes and the need for computationally efficient watermarking schemes for real-time applications motivate us to develop MC

watermarking schemes operating in the compressed domain.

3.2 Compressed Domain Video Watermarking

Many watermarking schemes, operating in the compressed/partially decompressed bit stream, have been proposed in the literature. The main motivation of such approaches is the reduced computational cost, which is important in the case of real-time applications. In this Section, a brief overview of the current video coding techniques is presented and then the main compressed domain watermarking approaches are investigated in the collusion-resistance perspective.

3.2.1 Video Coding Standards

The current video coding standards employ a *hybrid* coding strategy: a combination of predictive and transform coding, to effectively exploit the spatio-temporal redundancies present in the video sequences. Today's most widely used standard is the MPEG-2, which was standardized by the moving picture experperts group (MPEG) in 1994 [Tud95, L. 00]. In MPEG-2, the sequence to be coded is divided into *group of pictures* (GOP). The first frame in each GOP is the *intra-coded frame* (I-frame) which serves as the anchor frame for decoding. The I-frames are transform-coded without reference to other frames. The remaining frames in the GOP are the *inter-coded frames* which are hybrid-coded with reference to previously encoded frames. The first step in hybrid coding is the motion-compensated prediction in which a *current frame* is predicted from one or more neighboring *reference frames* on the basis of the estimated motion vectors. Depending on the nature of motion-compensation, there are two types of inter-coded frames: the *predictively coded frames* (P-frames) and the *bi-directionally predictive-coded frames* (B-frames). The I-frames and the P-frames are used as the prediction references for other P- and B-frames. The I-frames and the prediction-error frames corresponding to the P- and B-frames are subjected to the 8×8 block-DCT and then quantized. Finally, the quantized DCT coefficients are entropy-coded along with the motion vectors to generate the compressed stream.

The motion-compensated prediction structure in the current MPEG standards has many limitations, especially in the scalable coding applications. Motivated by the inherent scalability offered by the wavelet transform, the 3-D wavelet coding of video has recently emerged as a strong alternative to the hybrid coding approach [OdSW04]. In this approach, the motion-compensated prediction loop is replaced by the MC-TWT and the block-DCT by the 2-D wavelet transform. Depending on the order in which the temporal and spatial wavelet transforms are applied, there are two types of wavelet

video coding. In the 't+2D' approach, the MC-TWT is applied for temporally decorrelating the frames, followed by the 2D wavelet transform of the temporal sub-bands for spatial decorrelation. In the '2D+t' approach the order in which the transforms are applied is reversed: the 2-D wavelet transform followed by the MC-TWT of spatial sub-bands. The MPEG has started the exploration on 3-D wavelet transform based video coding and the 't+2D' structure has been accepted as the first working draft [R. 06].

3.2.2 Prior Work

In the pioneering work, Hartung et.al [HG98] proposed a watermarking scheme for MPEG-2 coded videos. The first step in the watermarking process is to partially decode the compressed stream to obtain the 8×8 block DCT coefficients of the I, B and P-frames. For each frame, a spread-spectrum watermark having the same dimensions as the frame is generated and then transformed to the 8×8 DCT. Finally, the DCT blocks of the watermark are added to the corresponding DCT blocks in the partially decoded stream to obtain the watermarked DCT blocks. These watermarked blocks are encoded along with the other parts such as the motion vectors and the header information, which are not altered by the watermarking process, to obtain the watermarked stream.

The authors have pointed out two problems associated with directly modifying the compressed stream. The first one, known as the watermark *drift*, arises due to the closed-loop prediction structure of the compression scheme. This can be better explained with the typical MPEG-2 GOP structure IBBPBBP$\cdots$. Consider the first four frames of the sequence. Any modifications in the I-frame will propagate to the P- and B-frames which use the I-frame as the prediction reference. In addition to this, if the P-frame is also modified by the watermarking process, the drift signal added to the B-frames will be the sum of the drift from the I-frame and that from the P-frame. Since the B-frames are not used as the prediction-reference, modifications to it will not propagate to other frames. However the P-frames are used as the prediction reference for future P- and B-frames, resulting in the accumulation of the drift signals towards the end of the GOP. For example, the drift from the I-frame and that from all the previous P-frames will contribute to the drift signal added to the last frame of the GOP. The watermark drift accumulation results in poor visual quality of the sequence and even degrades the detector performance.

To solve the drift problem, the authors proposed to add a *drift-compensation* signal to the inter-coded frames. For each inter-coded frame, the drift from its reference frames is first calculated and the drift-compensation signal is obtained by simply inverting the polarity of the drift signal. The second

problem addressed by the authors is the increase in the bit-rate due to watermark addition. They proposed a bit-rate control mechanism in which only selected non-zero DCT coefficients in each 8×8 block of the partially decompressed stream are modified by the watermarking process. The selection of the coefficients for watermark embedding is done in such a way that the number of bits required for encoding the watermarked sequence is less than or equal to that required for the uncompressed sequence.

A number of extensions of Hartung's method have been proposed. One essential component in all these methods is the drift-compensation module. In [ALC03], the authors propose a watermarking scheme for MPEG-4 coded sequences. The additional features in this method include synchronization templates to counter geometrical attacks, perceptual masking to improve the visual quality and the robustness, and an improved bit-rate control mechanism. More recently, two schemes have been proposed for watermarking of MPEG-2 coded videos [HS05, BDP05]; these are the adaptations of existing still-image watermarking techniques in the Hartung's framework.

One of the targeted applications of the compressed-domain watermarking schemes is the video-on-demand [STB+04], that requires real-time embedding of the watermark. In video-on-demand, a unique watermark corresponding to the user is added to each copy of the video. It is also expected that, in such an application, the watermarked sequence is likely to be subjected to many hostile attacks including the inter-frame collusion. However, to the best of the present author's knowledge, none of the compressed domain watermarking schemes are evaluated against the inter-frame collusion attacks. Analyzing the collusion-resistance properties of these schemes is rather straightforward. Since the drift-compensation prevents any temporal propagation of the watermark, these schemes essentially belong to the frame-by-frame watermarking approach. As we have shown in the previous chapter, this approach is vulnerable to inter-frame collusion attacks.

3.3 Proposed Approach

We propose a novel compressed-domain approach to the MC watermarking. The objectives of the proposed approach are to embed the watermark in the partially decompressed video and to exploit the available motion information in the compressed stream to reduce the computational cost. For this, we consider the video coded using the state-of-the-art motion-compensated block-DCT based and the emerging motion-compensated temporal filtering based techniques.

3.3.1 MC Watermarking of MPEG-2 Streams

The underlying idea behind the proposed MC watermarking scheme for MPEG-2 coded sequences is to exploit the drift signal, which is cancelled in the existing compressed domain watermarking techniques. We argue that instead of cancelling the drift signal, the proper usage of it will generate MC watermarks. In this scheme, the watermark needs to be embedded only in the I-frame of each GOP. In the intra-coded frames, MC watermarks are automatically added during decompression. This *drift-aided* MC watermarking scheme is first explained for three frames of a sequence and then extended for the complete sequence.

Consider three frames $\{\mathbf{x}_i, \quad i = 1, 2, 3\}$ of a host sequence. Let these frames be MPEG-2 coded as I, B and P-frames respectively. For simplifying the analysis, we make the following assumptions about the coding process:

1. The motion-estimation is done with integer-pixel accuracy
2. The effect of quantization is negligible.

The I, B and P-frames obtained by partial decoding can be now represented in the spatial-domain as

$$\begin{aligned}\mathbf{I}[\mathbf{n}] &= \mathbf{x}_1[\mathbf{n}] \\ \mathbf{P}[\mathbf{n}] &= \mathbf{x}_3[\mathbf{n}] - \mathbf{x}_1[\mathcal{M}_{1\to3}(\mathbf{n})] \\ \mathbf{B}[\mathbf{n}] &= \mathbf{x}_2[\mathbf{n}] - \frac{1}{2}\big(\mathbf{x}_1[\mathcal{M}_{1\to2}(\mathbf{n})] + \mathbf{x}_3[\mathcal{M}_{3\to2}(\mathbf{n})]\big)\end{aligned}$$

where $\mathcal{M}_{i\to j}$ is a motion-compensated mapping from frame i to frame j. Let a spread-spectrum watermark $\mathbf{w}$ be added to the I-frame in the spatial-domain and given by

$$\hat{\mathbf{I}}[\mathbf{n}] = \mathbf{I}[\mathbf{n}] + \mathbf{w}[\mathbf{n}]$$

where $\hat{\mathbf{I}}$ is the watermarked I-frame. When the sequence is decompressed, the watermarked frame $\mathbf{y}_1$ corresponding to the frame $\mathbf{x}_1$, under the above assumptions, is given by

$$\mathbf{y}_1[\mathbf{n}] = \hat{\mathbf{I}}[\mathbf{n}] = \mathbf{x}_1[\mathbf{n}] + \mathbf{w}[\mathbf{n}] \ . \tag{3.3.1}$$

For the frame $\mathbf{x}_3$ which is coded as a P-frame, the corresponding decompressed frame $\mathbf{y}_3$ can be expressed as

$$\mathbf{y}_3[\mathbf{n}] = \mathbf{P}[\mathbf{n}] + \mathbf{y}_1[\mathcal{M}_{1\to3}(\mathbf{n})] \ . \tag{3.3.2}$$

Substituting the value of $\mathbf{y}_1$ from Equation (3.3.1), we obtain

$$\begin{aligned}\mathbf{y}_3[\mathbf{n}] &= \mathbf{P}[\mathbf{n}] + \mathbf{x}_1[\mathcal{M}_{1\to3}(\mathbf{n})] + \mathbf{w}[\mathcal{M}_{1\to3}(\mathbf{n})] \\ &= \mathbf{x}_3[\mathbf{n}] + \mathbf{w}[\mathcal{M}_{1\to3}(\mathbf{n})] \end{aligned} \tag{3.3.3}$$

where $\mathbf{w}[\mathcal{M}_{1\to3}(\mathbf{n})]$ is the drift signal. If we consider a macro-block in $\mathbf{x}_3$, the drift signal added to it is the watermark corresponding to the best-matching block in frame $\mathbf{x}_1$. In other words, the addition of the drift signal during the decompression is equivalent to adding an MC watermark to frame $\mathbf{x}_3$.

The situation is slightly different for frame $\mathbf{x}_2$ which is coded as a B-frame. The corresponding decompressed frame $\mathbf{y}_2$ is given by

$$\begin{aligned}\mathbf{y}_2[\mathbf{n}] &= \mathbf{B}[\mathbf{n}] + \frac{1}{2}\big(\mathbf{y}_1[\mathcal{M}_{1\to2}(\mathbf{n})] + \mathbf{y}_3[\mathcal{M}_{3\to2}(\mathbf{n})]\big) \\ &= \mathbf{B}[\mathbf{n}] + \frac{1}{2}\big(\mathbf{x}_1[\mathcal{M}_{1\to2}(\mathbf{n})] + \mathbf{w}[\mathcal{M}_{1\to2}(\mathbf{n})]\big) \\ &\quad + \frac{1}{2}\big(\mathbf{x}_3[\mathcal{M}_{3\to2}(\mathbf{n})] + \mathbf{w}[\mathcal{M}_{1\to3}(\mathcal{M}_{3\to2}(\mathbf{n}))]\big) \\ &= \mathbf{x}_2[\mathbf{n}] + \frac{1}{2}\big(\mathbf{w}[\mathcal{M}_{1\to2}(\mathbf{n})] + \mathbf{w}[\mathcal{M}_{1\to3}(\mathcal{M}_{3\to2}(\mathbf{n}))]\big) \end{aligned} \tag{3.3.4}$$

where the drift signal $\frac{1}{2}\big(\mathbf{w}[\mathcal{M}_{1\to2}(\mathbf{n})] + \mathbf{w}[\mathcal{M}_{1\to3}(\mathcal{M}_{3\to2}(\mathbf{n}))]\big)$ is the watermark for frame $\mathbf{x}_2$. For a given macro block in $\mathbf{x}_2$, the watermark is the average of the watermarks in two matching blocks: one in frame $\mathbf{x}_1$ and the other in $\mathbf{x}_3$. If the motion-composition property

$$\mathcal{M}_{1\to3}(\mathcal{M}_{3\to2}(\mathbf{n})) = \mathcal{M}_{1\to2}(\mathbf{n})$$

holds for all points in the macro block, Equation (3.3.4) can be simplified as

$$\mathbf{y}_2[\mathbf{n}] = \mathbf{x}_2[\mathbf{n}] + \mathbf{w}[\mathcal{M}_{1\to2}(\mathbf{n})] \ . \tag{3.3.5}$$

In this case, the watermarks embedded in the macro block and its matching blocks are the same. From Equation(3.3.4) it is clear that, even if the motion-composition property does not hold, the watermark embedded in the macro block is correlated with those in the matching blocks.

Thus MC watermarks are generated for the P and B-frames from the embedded watermark in the I-frame during decompression.

Watermarking of complete sequence

Consider a typical MPEG-2 GOP structure $\mathbf{IBBPBBPBBPBBPBBI}\cdots$, with each GOP consisting of 12 frames. Continuing in the similar manner as above, we can show that the watermark embedded

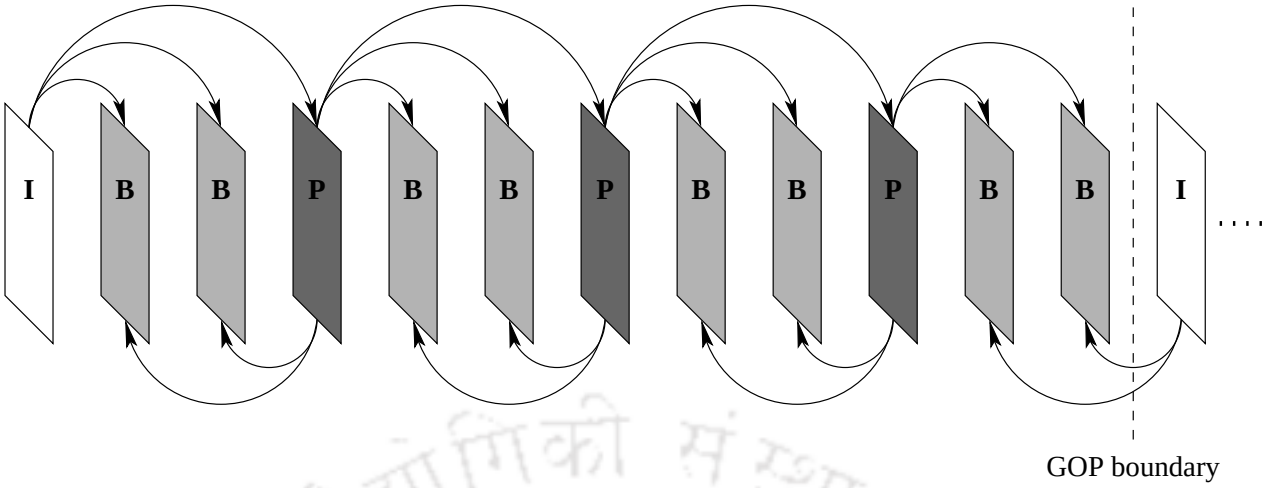


Figure 3.2: Propagation of the watermark in a GOP

in the I-frame of the first GOP will propagate till the last P-frame of the GOP. The watermarks in the last two B-frames of the GOP depend also on that embedded in the I-frame of next GOP. Now a watermark needs to be chosen for embedding in this I-frame. A simple way is to choose the watermark either as the same as or uncorrelated with that embedded in the previous I-frame. However this simple technique will make it vulnerable to inter-frame collusion, thereby defeating the very purpose of the watermarking scheme. An attacker with the knowledge of the embedding algorithm can obtain all the I-frames form the partially decoded stream, subject them to inter-frame collusion and finally replace the watermarked I-frames with the attacked ones. If the inter-frame collusion of the I-frames is successful, it will remove the watermark from all the frames. Moreover, such an embedding strategy will not maintain motion-coherency in the watermark across the GOP boundary.

Considering the above factors, we choose the motion-compensated prediction from the watermark in the last P-frame of the first GOP for embedding in the I-frame of the second GOP. In this way, the propagation of the watermark inside a GOP is extended across the GOP boundary. However, there are some difficulties associated with this approach. One difficulty is that the watermark of the P-frame is not available during embedding; it is generated only when the sequence is decompressed. This can be solved by exploiting the available motion vectors. From Equation (3.3.2), it is clear that the watermark in a P-frame can be calculated using the watermark in its reference frame and the corresponding motion vectors. So all we need to calculate the watermark in the last P-frame in a GOP are (1) the watermark in the I-frame and (2) the motion-vectors corresponding to the P-frames, which are readily available. Another difficulty is in obtaining the motion-compensated mapping between the last P-frame and the next I-frame since these motion trajectories are not estimated during the encoding process. This problem

can be solved by using the motion-inversion technique as shown by the dotted arrow in Figure 3.3. As

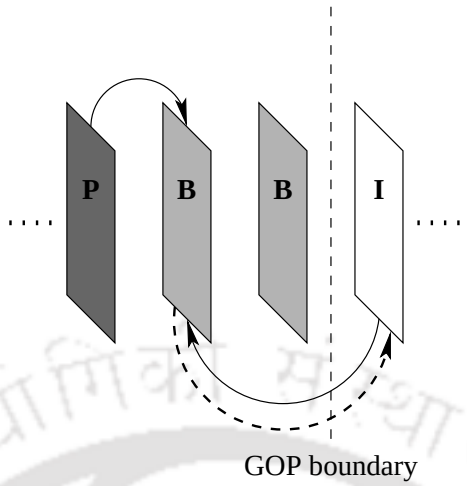


Figure 3.3: Propagation of watermark across GOP boundary

the motion vectors of the B-frame with respect to the I-frame is available, motion-inversion will give the motion vectors of the I-frame with respect to the B-frame. Now the motion composition property among the P-, B-, and I-frames could have been exploited to find the motion vectors of the I-frame with respect to the P-frame. However, due to motion-inversion, some points in the I-frame may not have correspondence with any point in the P-frame whereas some points may have multiple correspondence. The multiple correspondence problem can be handled by considering only one correspondence during the inversion process. The unconnected points correspond to the *newly-entered* areas in the I-frame. We add the watermark derived from the P-frame to the points where the correspondence exist and to the remaining unconnected points, the corresponding watermark in the same location in the first I-frame is added. Note that it is possible to add new watermark samples to the newly-entered areas. However, we choose the above strategy for simplicity.

The proposed watermarking scheme is performed in the following steps:

1. Partially decode the sequence to obtain the I-frame and the motion vectors in the first GOP
2. Add a watermark to the I-frame
3. Calculate the watermark that will propagate to the last P frame of the current GOP
4. Decode the I-frame and the motion vectors in the next GOP
5. Calculate the motion vectors of the I frame with respect to the last P-frame of the previous GOP

6 Derive the watermark for the I-frame from that obtained in step 3 and add to the I-frame. In the unconnected points, add the watermark corresponding to the same location in the I-frame of first GOP

7 Continue steps $3-6$ until the scene boundary is reached

3.3.2 MC-TWT Domain Watermarking

We now propose a motion-coherent watermarking scheme for the sequences coded using the emerging MC-TWT based scheme. In this method the watermark is added to the low-pass temporal wavelet frames obtained by partial decoding of the coded sequence. The working of the watermarking scheme is explained with the Haar wavelet. Like in the MPEG-2 watermarking scheme, it is assumed that the motion-estimation is with integer-pixel accuracy and the quantization effect is negligible.

Suppose a video sequence $\{\mathbf{x}_k[\mathbf{n}],\ k = 0, 1, \cdots N_f - 1\}$ is decomposed up to the j-th level using the lifting based MC-TWT. The corresponding low-pass and high-pass temporal frames are given by the iterative steps

$$\left.\begin{aligned} \mathbf{h}_k^i[\mathbf{n}] &= \mathbf{l}_{2k+1}^{i-1}[\mathbf{n}] - \mathbf{l}_{2k}^{i-1}[\mathcal{M}_{(2k)2^{i-1}\rightarrow(2k+1)2^{i-1}}(\mathbf{n})] \\ \mathbf{l}_k^i[\mathbf{n}] &= \mathbf{l}_{2k}^{i-1}[\mathbf{n}] + \tfrac{1}{2}\mathbf{h}_k^i[\mathcal{M}_{(2k+1)2^{i-1}\rightarrow(2k)2^{i-1}}(\mathbf{n})] \end{aligned}\right\}, \quad k = 0, 1, \cdots, (N_f/2^i) - 1 \tag{3.3.6}$$

with $\mathbf{l}_k^0[\mathbf{n}] = \mathbf{x}_k[\mathbf{n}]$ and $i = 1, 2, \cdots, j$. Similarly, the corresponding reconstruction lifting steps are given by

$$\left.\begin{aligned} \mathbf{l}_{2k}^i[\mathbf{n}] &= \mathbf{l}_k^{i+1}[\mathbf{n}] - \tfrac{1}{2}\mathbf{h}_k^{i+1}[\mathcal{M}_{(2k+1)2^i\rightarrow(2k)2^i}(\mathbf{n})] \\ \mathbf{l}_{2k+1}^i[\mathbf{n}] &= \mathbf{h}_k^{i+1}[\mathbf{n}] + \mathbf{l}_{2k}^i[\mathcal{M}_{(2k)2^i\rightarrow(2k+1)2^i}(\mathbf{n})] \end{aligned}\right\}, \quad k = 0, 1, \cdots, (N_f/2^{i+1}) - 1 \tag{3.3.7}$$

where $i = j-1, j-2, \cdots, 0$. Suppose a spread-spectrum watermark $\{\mathbf{w}_k[\mathbf{n}],\ k = 0, 1, \cdots N_f/2 - 1\}$ is added to the low-pass temporal frames resulting from the first level of MC-TWT decomposition. The watermarked low-pass frame at this level is given by

$$\hat{\mathbf{l}}_k^1[\mathbf{n}] = \mathbf{l}_k^1[\mathbf{n}] + \mathbf{w}_k[\mathbf{n}], \qquad k = 0, 1, \cdots, (N_f/2) - 1\ . \tag{3.3.8}$$

The watermarked frames $\{\mathbf{y}_k[\mathbf{n}] = \hat{\mathbf{l}}_k^0[\mathbf{n}], \quad k = 0, 1, \cdots N_f - 1\}$ are obtained by using the reconstruction steps (3.3.7) as

$$\begin{aligned} \mathbf{y}_{2k}[\mathbf{n}] &= \mathbf{x}_{2k}[\mathbf{n}] + \mathbf{w}_k[\mathbf{n}] \\ \mathbf{y}_{2k+1}[\mathbf{n}] &= \left(\mathbf{h}_k^1[\mathbf{n}] + \mathbf{x}_{2k}[\mathcal{M}_{2k\rightarrow2k+1}(\mathbf{n})]\right) + \mathbf{w}_k[\mathcal{M}_{2k\rightarrow2k+1}(\mathbf{n})] \\ &= \mathbf{x}_{2k+1}[\mathbf{n}] + \mathbf{w}_k[\mathcal{M}_{2k\rightarrow2k+1}(\mathbf{n})]\ . \end{aligned}$$

Similarly, if the watermark is added to the low-pass temporal frames of the second level of MC-TWT decomposition, the resulting watermarked frames are

$$\mathbf{y}_{2k}[\mathbf{n}] = \begin{cases} \mathbf{x}_{2k}[\mathbf{n}] + \mathbf{w}_{(k/2)}[\mathbf{n}], & k = 0, 2, \cdots, (N_f/2) - 2 \\ \mathbf{x}_{2k}[\mathbf{n}] + \mathbf{w}_{(k-1)/2}[\mathcal{M}_{(k-1)2 \to 2k}(\mathbf{n})], & k = 1, 3, \cdots, (N_f/2) - 1 \end{cases} \tag{3.3.9}$$

$$\mathbf{y}_{2k+1}[\mathbf{n}] = \begin{cases} \mathbf{x}_{2k+1}[\mathbf{n}] + \mathbf{w}_{(k/2)}[\mathcal{M}_{2k \to 2k+1}(\mathbf{n})], & k = 0, 2, \cdots, (N_f/2) - 2 \\ \mathbf{x}_{2k+1}[\mathbf{n}] + \mathbf{w}_{(k-1)/2}[\mathcal{M}_{(k-1)2 \to 2k}(\mathcal{M}_{2k \to 2k+1}(\mathbf{n}))], & k = 1, 3, \cdots, (N_f/2) - 1 \end{cases} . \tag{3.3.10}$$

Under the assumption of the composition of motion operators, i.e., $\mathcal{M}_{k_1 \to k_2}(\mathcal{M}_{k_2 \to k_3}(\mathbf{n})) = \mathcal{M}_{k_1 \to k_3}(\mathbf{n})$, it can be shown that watermarking in the low-pass frames of the $L - th$ level MC-TWT decomposition results in

$$\mathbf{y}_{2^L k+i}[\mathbf{n}] = \begin{cases} \mathbf{x}_{2^L k+i}[\mathbf{n}] + \mathbf{w}_k[\mathbf{n}], & i = 0 \\ \mathbf{x}_{2^L k+i}[\mathbf{n}] + \mathbf{w}_k[\mathcal{M}_{(2^L k) \to 2^L k+i}(\mathbf{n})], & i = 1, 2, \cdots, 2^L - 1 \end{cases} \tag{3.3.11}$$

where $k = 0, 1, \cdots, (N_f/2^L) - 1$. It can be further shown that the first relation in Equation (3.3.11) holds irrespective of the choice of the wavelet filter. But the second relation is specific for the Haar wavelet; for other wavelets, corresponding relations can be derived. It is also clear that the watermarks in the frames given by the second expression of Equation (3.3.11) are coherent with motion. For the MC-TWT domain watermarking to be motion-coherent, the watermark frames $\{\mathbf{w}_k[\mathbf{n}], \quad k = 0, 1, \cdots, (N_f/2^L) - 1\}$ are to be made motion-coherent.

Note that the motion coherency in the watermark frames $\{\mathbf{w}_{2^L k}[\mathbf{n}], \quad k = 0, 1, \cdots, (N_f/2^L) - 1\}$ is important for the security of the watermark. If the low-pass frames are marked with redundant watermarks $\big(\mathbf{w}_k[\mathbf{n}] = \mathbf{w}[\mathbf{n}], \quad k = 0, 1, \cdots, (N_f/2^L) - 1\big)$, an attacker having the knowledge of the embedding algorithm can group the frames $\{\mathbf{y}_{2^L k}[\mathbf{n}], \quad k = 0, 1, \cdots, (N_f/2^L) - 1\}$ and apply the WER attack to get an estimate of the watermark frame $\mathbf{w}[\mathbf{n}]$. Embedding statistically independent watermarks in these frames is also not secure. Such watermarks embedded in the frames from a static scene can be removed using the FTF attack. Thus the watermark frames $\{\mathbf{w}_k[\mathbf{n}], \quad k = 0, 1, \cdots, (N_f/2^L) - 1\}$, designed according to the motion content, are given by

$$\mathbf{w}_k[\mathbf{n}] = \begin{cases} \mathbf{w}[\mathbf{n}], & k = 0 \\ \mathbf{w}_{k-1}[\mathcal{M}_{2^L(k-1) \to 2^L k}(\mathbf{n})], & k = 1, 2 \cdots, (N_f/2^L) - 1 \end{cases} \tag{3.3.12}$$

The proposed watermarking technique is summarized in the following steps.

1. Segment the video frames into scenes

2. Estimate the motion trajectories using a motion model of choice

3 Apply the lifting based MC-TWT to frames in each scene using a suitable filter

4 Mark the resulting low-pass temporal frames with spread spectrum watermarks generated using (3.3.12)

5 Apply the inverse MC-TWT to get the watermarked frames

Note that we have presented the above algorithm assuming uncompressed video. If the MC-TWT coded video is available, the temporal low-pass frames can be obtained by partial decompression. The resulting low-pass frames can then be marked with spread-spectrum watermarks using the steps 4 and 5 above.

3.4 Performance Analysis

In this section, the performance of the proposed watermarking schemes is analyzed in terms of the coding parameters, motion-coherency, and the resistance to known inter-frame collusion attacks. The analysis is carried out only for the MPEG-2 watermarking scheme. However, the analysis can be extended for the MC-TWT based watermarking scheme without the loss of any generality.

3.4.1 Coding Parameters

The coding parameters that affect the performance of proposed watermarking schemes are the *accuracy of motion-estimation* and *bit-rate*.

Accuracy of Motion-Estimation

In the derivation of the proposed watermarking schemes, it is assumed that the motion-estimation during the compression process is with integer-pixel accuracy. But the current video coding standards support sub-pixel accurate motion-estimation. For example, the MPEG-2 standard supports up to half-pixel accuracy. The sub-pixel accurate motion-compensation improves the compression efficiency by reducing the energy of the motion-compensated prediction error frames. However, the spatial interpolation associated with the sub-pixel motion-compensation affects the embedded watermarks in two ways: attenuates the energy and introduces spatial low-pass components.

The watermark embedded in the I-frame of the first GOP is not affected by the interpolation. But the watermarks propagated to the P- and B- frames undergo interpolation. As a result of multiple interpolation, towards the end of the GOP, the attenuation of the propagated watermark increases and more

and more low-pass components will be introduced into the watermark. Since the propagation of the watermark is extended across the GOP boundaries, after a certain number of GOPs, the watermark energy reduces significantly and the watermark may become visible due to the low-frequency components.

The attenuation of the watermark can be reduced by *boosting* the watermark before adding it to the I-frames. However this will not solve the introduction of low-frequency components. Another solution is to terminate the propagation of the watermark after a certain number of GOPs and then start with a new watermark in the next GOP. In this case the motion-coherency in the watermark will be lost across the GOPs where the propagation of the watermark is terminated and a new watermark is embedded. Note that the compression standards only specifies the maximum allowable precision of motion-vectors and is not necessarily be used. So, the motion-estimation in the compression process may be limited to the integer-pixel accuracy. In this case, the compression efficiency is compromised for avoiding the interpolation of the embedded watermarks.

Bit-Rate

In compressed-domain watermarking, it is desirable that the watermark embedding should not increase the bit-rate beyond a certain range. The compressed-domain watermarking schemes employ a rate control mechanism to keep the bit-rate of the watermarked sequence within the allowable range. The rate control is achieved by compromising the detectability of the watermark [HG98] or by trading the perceptual quality of the watermarked sequence [ALC03]. Any of these bit-rate control methods can be used in the proposed watermarking schemes, if necessary. Here we analyze the impact of the bit-rate control on the proposed watermarking schemes in comparison with other compressed domain watermarking methods.

The impact of rate control on the embedded watermark depends on the increase in bit-rate requirements due to watermark addition. In the conventional compressed-domain watermarking schemes, the increase in the bit-rate is due to (1) the watermark added to both the intra-coded and inter-coded frames and (2) the drift compensation signal added to the inter-coded frames. On the other hand, in the proposed MPEG-2 watermarking scheme, the watermark is added only to the intra-coded frames and there is no drift-compensation. So, the impact of rate control is negligible in the proposed scheme as compared to other compressed domain watermarking schemes. It should be noted that the watermark added to the I-frames will be subjected to quantization, even if the bit-rate control is not employed. Due to fine quantization in the I-frames, its impact on the watermark can be neglected. The quantized watermark will propagate to the inter-coded frames and hence the motion-coherency in the watermark

is not affected by the quantization.

The *bit-saving* in the proposed watermarking scheme is achieved in any motion-coherent watermarking method as well. To make this generalization, let us consider the MPEG-2 encoding of a watermarked sequence. Watermark embedding increases the number of bits required to code the I-frames and this increase is independent of the motion-coherency in the watermark. But the situation is different for the inter-coded P- and B-frames. If the watermark is motion-coherent, the watermarks in the P- and B-frames can be predicted by motion-compensation. So the number of bits required to code the watermarked P- and B-frames are the same as that required for the corresponding host frames. On the other hand, if the watermark is not motion-coherent, the watermark in the P- and B-frames are predicted from uncorrelated watermarks in the corresponding reference frames. The prediction from uncorrelated reference frames increases the variance of the prediction error frame and more bits are required to code the prediction error frames. Note that in other compressed-domain watermarking schemes, the increased variance of the prediction error frames is introduced by the drift-compensation signal.

3.4.2 Motion-Coherency in the Watermark

An *ideal motion-coherent* watermark, as the name implies, should be coherent with the motion in the video scene. That is, whenever a 3-D point is projected in the frame, it should carry the same watermark sample. However, in practical cases, it is difficult to achieve this level of motion-coherency due to the complex nature of the motion in natural videos. All the MC watermarking schemes first *estimate* the motion in the scenes using a suitable estimation algorithm and then embed a watermark coherent with the *estimated* motion. So, the watermark generated by these schemes are coherent with the estimated motion. In all the motion-estimation techniques, a motion model is assumed and the model parameters are then estimated. For example, in the mosaic-based approach, the camera motion is modelled with a first-order polynomial model where as in the proposed watermarking schemes, a simple block-based translational motion model is used to take account of both the camera and object motion. None of these motion models can capture the ‘true motion’ in real videos and hence the achieved motion-coherency is generally far from the ideal. However, the main objective of the MC watermarking is to counter the MC-FTF attack, not to generate an ideal MC watermark. In the MC-FTF attack, the attacker estimates the motion and the frames are averaged along the estimated motion-trajectories. The success of an MC watermarking scheme in countering the MC-FTF attack depends on the model used by the

attacker to estimate the motion. It is shown in the previous chapter that the MC-FTF attack using block-based motion estimation is effective in removing the watermark generated by many existing watermarking schemes. The proposed MC watermarking schemes, which embed coherent watermark along the motion-trajectories estimated using the block-based motion model, are expected to counter such attack.

In a video sequence, it might happen that some areas disappear from a frame and reappear in a later frame. This situation arises in the following cases:

1. Due to object motion, a portion of the background or another object is occluded in a frame and reappears in a later frame.

2. Due to camera motion, some areas in the background or moving objects may move out of the field of view and re-enters at a later time.

Let us now analyze how the MC watermarking schemes handle such areas. In the mosaic-based watermarking scheme with perfect frame registration, such areas will not cause a problem since the similar background areas in all frames are aligned before watermark embedding. In other words, an area in the background that re-enters the sequence after a certain number of frames will carry the same watermark as in its previous occurrences. The proposed MPEG-2 watermarking scheme handles such areas in a different manner. Consider a macro block in a P-frame which has not appeared in its reference frame. The watermark for this block is derived from its best-matching block in the reference frame, without considering the watermark in its previous appearances. As a result, the motion-coherency in the watermark is lost in such blocks. Likewise, such blocks in the intermediate I-frames also lose motion-coherency in the watermark. However this problem does not exist for a B-frame. It is unlikely that a block appears only in a B-frame, not in either of its prediction references.

From the security point of view, the lack of motion-coherency creates two problems: (1) the same areas separated by a number of frames may carry uncorrelated watermarks and (2) dissimilar areas may carry the same watermark. In the first case, the MC-FTF attack may be employed to remove the watermark and the in the second case, the WER attack may be employed in the block-level. The efficiency of such attacks in removing the watermark depends on the size of the newly entered areas. Furthermore, in the case of the MC-FTF attack, the attacker chooses only a small number of frames in a temporal window. The frame in which an area is disappears and that in which it re-enters are generally separated by a large number of frames. So it is unlikely that both the frames are included in the same temporal window for the MC-FTF attack. In summary, even though the motion-coherency

in the watermark is lost in the newly entered areas, we believe that it is not critical from the security perspective.

3.4.3 Improved Robustness to Compression

An issue similar to the bit-rate control is the robustness of the watermark when the sequence is re-encoded at a lower bit-rate. The re-encoding could be aimed at reducing the storage space and transmission bandwidth or an intentional attempt to remove the watermark. Let us analyze how different watermarking schemes perform when the watermarked frames subjected to re-encoding. We generalize the analysis by comparing the performance of the motion-coherent and motion-incoherent watermarking schemes when subjected to MPEG-2 re-encoding.

The effect of re-encoding on the embedded watermark is that it is subjected to quantization with a larger quantizer step size which depends on the bit-rate. The effect of quantization on watermark detector performance has been extensively studied in literature [EG01, XL01, BS04]. In general, the quantization effect on the watermark can be considered as the addition of a noise term. The additive noise is uncorrelated with the watermark for small quantizer step sizes and is negatively correlated with the watermark for larger step sizes [EG01]. In other words, for large step size, the additive quantization noise reduces the watermark detector's performance and hence can be considered as a *watermark removal* operation. As mentioned earlier, the prediction error frames corresponding to the P- and B- frames are quantized coarsely as compared to the I-frames. So the impact of re-encoding on the embedded watermark depends on the contribution of the watermark in the prediction error frames. If the watermark is motion-coherent, no watermark component is introduced to the prediction error frames. On the other hand, if the watermark is not coherent with motion, the prediction error frames contain contributions from the watermark, which is subjected to coarse quantization. As a result, the motion-coherent watermark is expected to possess better robustness to re-encoding as compared to the motion-incoherent watermark.

3.4.4 Robustness to Known Inter-Frame Collusion Attacks

It is obvious that the motion-coherent watermarking offers good robustness against MC-FTF attack. Also, it is straightforward to show that it is robust to other known inter-frame collusion attacks as well. As explained in the previous chapter, Type-1 collusion is possible when visually dissimilar frames are embedded with highly correlated watermarks and Type-2 collusion is effective in frames from a static

scene carrying uncorrelated watermarks. The correlation between the watermarks embedded in different frames by a motion-coherent watermarking scheme depends on the motion between the frames. For a static scene, the scheme embeds highly correlated watermarks in different frames and hence survives the Type-1 collusion. On the other hand, if the frames are from a dynamic scene, the embedded watermarks are uncorrelated with each other. So, Type-2 collusion will not be effective. In other words, the motion-coherency in the watermark is a *sufficient* condition to guarantee robustness against all the known inter-frame collusion attacks. This important property of motion-coherent watermark is verified experimentally in the following Section.

3.5 Experimental Results

This Section details the results of the experimental studies performed to evaluate the performance of the proposed watermarking schemes. Since our objective is to evaluate the performance of the proposed watermarking schemes against the MC-FTF and other known inter-frame collusion attacks, we have not considered the quantization step in the compression process for simplicity. All the experiments are conducted on uncompressed video sequences by *emulating* the partial decoding of the coded sequences. For the MPEG-2 watermarking scheme, the motion vectors and the prediction error frames are computed directly from the uncompressed sequences. Similarly, the motion vectors and the temporal low-pass and high-pass frames are computed from the uncompressed sequences for the MC-TWT domain watermarking scheme. Note that the quantization is the only lossy operation in the coding process. So, these implementations are equivalent to watermarking of the corresponding partially decoded sequences if the quantization effect is neglected.

A number of test video sequences, including those considered in the previous chapter to evaluate the performance of the MC-FTF attack are considered in the experiments. The first 64 frames from each of the sequences are used in the experiments. For the MPEG-2 watermarking, the sequences are divided into a typical MPEGG-2 GOP structure IBBPBBPBBPBB with a GOP length of 12 frames. The motion vectors for the inter-coded frames are computed using the FSBM with integer-pixel accuracy and a macro-block size of 16×16 pixels. In the case of the MC-TWT domain watermarking, the Haar wavelet and 3 level temporal decomposition are considered. The motion vectors are estimated using the HVSBM with integer-pixel accuracy. The embedding strength of the watermarks in both the cases is chosen such that the PSNR of the watermarked sequences is around 38 dB. The average NC values and the PSNR of the attacked sequences are used as the performance measures.

Figures 3.4 and 3.5 show the sample frames from the *Foreman* and *Antibes* sequences marked using the MPEG-2 watermarking scheme and the corresponding watermarks. The frames are the last frame of the respective sequences. Note that there is no visible artifact in the *Foreman* sequence. However, some artifacts are visible in the top left region of the *Antibes*. The *Antibes* is a synthetic sequence with uniform horizontal translational motion from left to right. So the left portion in each frame contains *newly-entered* areas, i.e., they are not present in the previous frame. The watermark embedded in these areas depends on the motion vectors estimated from the previous frame. It is observed that the estimated motion vectors are *zero* for the top-left porion of each frame, which is a *smooth* area. This may be the reason for the spatial correlation in the watermark in those areas. On the other hand, in the bottom-left portion, the newly-entered area contains spatial high-frequency components and the motion-vectors change from frame to frame. Hence no spatial-correlation is introduced in the watermark.

3.5.1 Performance Against MC-FTF Attack

The first set of experiments evaluates the performance of the proposed watermarking schemes against the MC-FTF attack. The MC-FTF attack with the Haar wavelet and up to 3 level MC-RTWT decomposition is considered. Even though the integer-pixel motion estimation is used in the watermark embedding process, the attacker is free to choose any precision for the motion estimation. It is also possible that the attacker employs a motion estimation technique, other than the one used for watermark embedding. We evaluate the performance of the watermarking schemes considering a few such cases.

Table 3.1 depicts the performance of the MPEG-2 watermarking scheme when the attacker uses the same motion estimation algorithm as in the watermark embedding. It is observed that the watermarking scheme offers a good degree of robustness to the attacks, even for the attack with sub-pixel accurate motion estimation. For all precisions of motion-estimation, except for the *Antibes*, a decrease in the NC performance is observed with an increase in the level of MC-RTWT decomposition. The block-based motion estimation captures the translational motion in the Antibes sequence perfectly and hence the motion-composition property is satisfied. Other sequences contain non-translational motion and the estimated motion vectors do not hold the composition property. This explains the decrease in the NC values with the increasing level of the MC-RTWT decomposition. Another point to be noted is the change in the NC values with varying the precision of motion-estimation used in the attack. As explained in the previous chapter, the spatial-interpolation involved in sub-pixel accurate motion-estimation decreases the attack performance (high NC values). The Antibes sequence which contains

only integer-pixel motion is not affected by the sub-pixel motion estimation. But for other sequences, the NC values increase with the precision of motion-estimation.

On comparing the NC values given in Table 3.1(a) and Table 3.1(b), it is evident that the MPEG-2 watermarking scheme offers better robustness to MC-FTF attack when the motion vectors are estimated from the watermarked sequence. This improved robustness is attributed to the change in the estimated motion vectors due to watermark embedding. It is well known that the motion-vectors estimated using the block-based motion from the frames containing non-translational motion is non-invertible. The embedded watermark forces the motion estimation algorithm to follow the motion-trajectories which carry correlated watermarks, there by reducing the motion-inversion error. For the Antibes sequence which contains only translational motion, there will not be any change in the motion vectors (except for the few blocks in the occluded/uncovered areas). However, other sequences contain non-translational motion and the bias in the estimated motion-vectors towards the motion-trajectories used for watermark embedding results in the improved robustness.

The NC performance of the MPEG-2 watermarking scheme against the MC-FTF attack with a motion estimation scheme other than the one used in watermark embedding is reported in Table 3.2. On comparing with the results presented in Table 3.1, it can be observed that the change in the motion estimation algorithm marginally decreases the performance against the MC-FTF attack. This is due to the difference in the motion vectors estimated using the two techniques. Note that there is almost no change in the performance for the Antibes sequence, but the change is prominent for the *Mobile* and the *Stefan* sequences, which contain non-translational motion. A comparative study on the performance of the MPEG-2 watermarking scheme with the SS and the SS-1 embedding schemes against the MC-FTF attack is carried out and the results are tabulated in Table 3.3. Significantly better robustness of the MPEG-2 watermarking scheme is evident from these results.

The same set of experiments is carried out to evaluate the performance of the MC-TWT domain watermarking scheme against the MC-FTF attack. Table 3.4 reports the performance when the same motion-estimation algorithm is used for both the embedding and the attacking. From these results and the results from other experiments, it is observed that the performance of the MC-TWT domain watermarking scheme is comparable with that of the MPEG-2 watermarking scheme when subjected to the MC-FTF attack.

3.5.2 Performance Against WER and FTF Attacks

The next set of experiments is to evaluate the performance of the proposed watermarking schemes against the WER and the FTF attacks. The performance is compared with that of two frame-by-frame additive watermarking schemes: the SS-1 and the SS embedding. In all the schemes, the embedding strength was chosen in such a way that the PSNR of the watermarked sequence is around 38 dB. The PSNR of the attacked sequence with respect to the host sequence is used as the measure of the visual quality of the attacked video.

For the WER attack, the watermarked video frames are spatially low-pass filtered using an adaptive Wiener filter with a 5×5 window [VPH$^+$00]. The Wiener filter is implemented using the MATLAB function. The resulting low-pass frames are subtracted from the watermarked frames and the differences are averaged to get an estimate of the watermark. Finally, the attacked frames are obtained by remodulating the watermarked frames with the estimated watermark [DD04b]. The performance of the watermarking schemes against the WER attack is presented in Table 3.5. As expected, the attack is effective against the SS-1 watermarking scheme while the other schemes survive it. In the case of the SS-1 watermarking, the PSNR of the attacked videos is higher than that of the watermarked sequences (38 dB) and the increase in the PSNR is at the cost of the decrease in the NC values. As explained in the previous Section, the robustness of the proposed watermarking schemes against the WER attack owes to the embedding of dynamic watermarks in the dynamic frames.

In the experiments to evaluate the performance against the FTF attack, we consider two sequences: *News* and *Akiyo*, in addition to the sequences used in the previous experiments. These sequences are selected due to their *near-static* nature, for which the FTF attack is known to be effective. A temporal window width of 9 frames is used in the experiments. From the results presented in Table 3.6 it can be observed that the SS-1 watermarking scheme survives the FTF attack in both the static and dynamic sequences whereas the detectability of the SS watermark is considerably degraded. On the other hand, the watermark embedded using the proposed schemes survives the FTF attack in the static sequences and becomes undetectable in the dynamic sequences. Note that the poor detectability of the watermark in the dynamic sequences is not critical since the quality of the attacked videos is significantly degraded as indicated by the low PSNR values. The poor visual quality of the attacked sequences is evident from the sample frames shown in Figure 3.6.

Sequence	integer-pixel			half-pixel			quarter-pixel		
	Level 1	Level 2	Level 3	Level 1	Level 2	Level 3	Level 1	Level 2	Level 3
Antibes	0.98	0.96	0.95	0.98	0.97	0.96	0.98	0.97	0.96
Foreman	0.86	0.72	0.56	0.9	0.8	0.68	0.9	0.82	0.7
Coastguard	0.92	0.83	0.69	0.94	0.87	0.78	0.95	0.9	0.81
Mobile	0.84	0.77	0.63	0.91	0.85	0.76	0.91	0.86	0.77
Stefan	0.87	0.74	0.57	0.9	0.8	0.68	0.91	0.83	0.71

(a)

Sequence	integer-pixel			half-pixel			quarter-pixel		
	Level 1	Level 2	Level 3	Level 1	Level 2	Level 3	Level 1	Level 2	Level 3
Antibes	0.98	0.97	0.95	0.98	0.97	0.96	0.99	0.98	0.97
Foreman	0.89	0.77	0.63	0.92	0.84	0.74	0.94	0.87	0.78
Coastguard	0.93	0.85	0.74	0.95	0.9	0.82	0.96	0.92	0.86
Mobile	0.84	0.78	0.64	0.92	0.86	0.77	0.92	0.88	0.79
Stefan	0.88	0.77	0.61	0.92	0.83	0.72	0.93	0.86	0.75

(b)

Table 3.1: Detector performance of the proposed MPEG-2 watermarking scheme against the MC-FTF attack with varying levels of MC-RTWT decomposition and varying precision of motion-estimation. The motion vectors for the attack are estimated using the FSBM from (a) the host sequence and (b) the watermarked sequence.

Sequence	integer-pixel			half-pixel			quarter-pixel		
	Level 1	Level 2	Level 3	Level 1	Level 2	Level 3	Level 1	Level 2	Level 3
Antibes	0.97	0.96	0.95	0.98	0.97	0.96	0.98	0.97	0.96
Foreman	0.8	0.64	0.48	0.86	0.74	0.61	0.87	0.75	0.63
Coastguard	0.9	0.79	0.66	0.92	0.85	0.75	0.93	0.87	0.77
Mobile	0.83	0.75	0.61	0.91	0.84	0.74	0.91	0.86	0.76
Stefan	0.83	0.69	0.53	0.88	0.76	0.64	0.88	0.78	0.66

(a)

Sequence	integer-pixel			half-pixel			quarter-pixel		
	Level 1	Level 2	Level 3	Level 1	Level 2	Level 3	Level 1	Level 2	Level 3
Antibes	0.98	0.97	0.96	0.99	0.98	0.97	0.99	0.98	0.97
Foreman	0.85	0.71	0.57	0.9	0.81	0.7	0.91	0.83	0.73
Coastguard	0.92	0.83	0.71	0.94	0.88	0.79	0.95	0.9	0.82
Mobile	0.85	0.77	0.63	0.92	0.86	0.77	0.92	0.88	0.79
Stefan	0.86	0.73	0.58	0.91	0.81	0.7	0.92	0.83	0.72

(b)

Table 3.2: Detector performance of the proposed MPEG-2 watermarking scheme against the MC-FTF attack with varying levels of MC-RTWT decomposition and varying precision of motion-estimation. The motion vectors for the attack are estimated using the HVSBM from (a) the host sequence and (b) the watermarked sequence.

Sequence	SS-1		SS		MPEG-2	
	FSBM	HVSBM	FSBM	HVSBM	FSBM	HSVM
Antibes	0.39	0.25	0.14	0.14	0.95	0.96
Foreman	0.38	0.41	0.17	0.18	0.63	0.57
Coastguard	0.29	0.25	0.16	0.16	0.74	0.71
Mobile	0.19	0.21	0.14	0.14	0.64	0.63
Stefan	0.42	0.4	0.18	0.18	0.61	0.58

Table 3.3: Comparative performance of the proposed MPEG-2 watermarking scheme against the MC-FTF attack with 3 level MC-RTWT decomposition. The motion vectors for the attack are estimated from the watermarked sequence by using the FSBM and the HSVBM with an integer-pixel accuracy.

Sequence	integer-pixel			half-pixel			quarter-pixel		
	Level 1	Level 2	Level 3	Level 1	Level 2	Level 3	Level 1	Level 2	Level 3
Antibes	0.97	0.95	0.93	0.98	0.96	0.94	0.98	0.96	0.94
Foreman	0.86	0.68	0.53	0.89	0.77	0.65	0.9	0.79	0.68
Coastguard	0.9	0.75	0.6	0.92	0.8	0.69	0.93	0.83	0.72
Mobile	0.84	0.71	0.59	0.9	0.81	0.71	0.91	0.82	0.73
Stefan	0.85	0.68	0.52	0.89	0.77	0.65	0.9	0.79	0.67

Table 3.4: Detector performance of the proposed MC-TWT domain watermarking scheme against the MC-FTF attack with varying levels of MC-RTWT decomposition and varying precision of motion-estimation. The motion vectors for the attack is estimated from the watermarked sequence by using the HVSBM.

Sequence	SS-1		SS		MPEG-2		MC-TWT	
	PSNR	NC	PSNR	NC	PSNR	NC	PSNR	NC
Antibes	43.13	0.18	35.67	0.98	35.77	0.97	35.69	0.97
Foreman	43.91	0.14	35.68	0.97	36.17	0.94	35.73	0.96
Coastguard	39.08	0.45	35.62	0.99	35.89	0.98	35.66	0.98
Mobile	38.2	0.55	35.61	0.99	35.85	0.98	35.62	0.99
Stefan	38.71	0.49	35.62	0.99	36.1	0.97	35.7	0.97

Table 3.5: Comparative performance of the proposed watermarking schemes against the WER attack.

Sequence	SS-1		SS		MPEG-2		MC-TWT	
	PSNR	NC	PSNR	NC	PSNR	NC	PSNR	NC
Antibes	26.11	1.00	26.33	0.11	26.33	0.14	26.33	0.13
Foreman	26.43	1.00	26.67	0.11	26.65	0.19	26.65	0.19
Coastguard	25.3	1.00	25.48	0.11	25.47	0.21	25.47	0.16
Mobile	20.26	1.00	20.32	0.11	20.32	0.24	20.32	0.17
Stefan	20.72	1.00	20.79	0.11	20.78	0.28	20.78	0.26
News	35.79	1.00	38.54	0.11	35.96	0.95	36.00	0.93
Akiyo	31.86	1.00	32.77	0.11	32.04	0.86	32.05	0.82

Table 3.6: Comparative performance of the proposed watermarking schemes against the FTF attack.

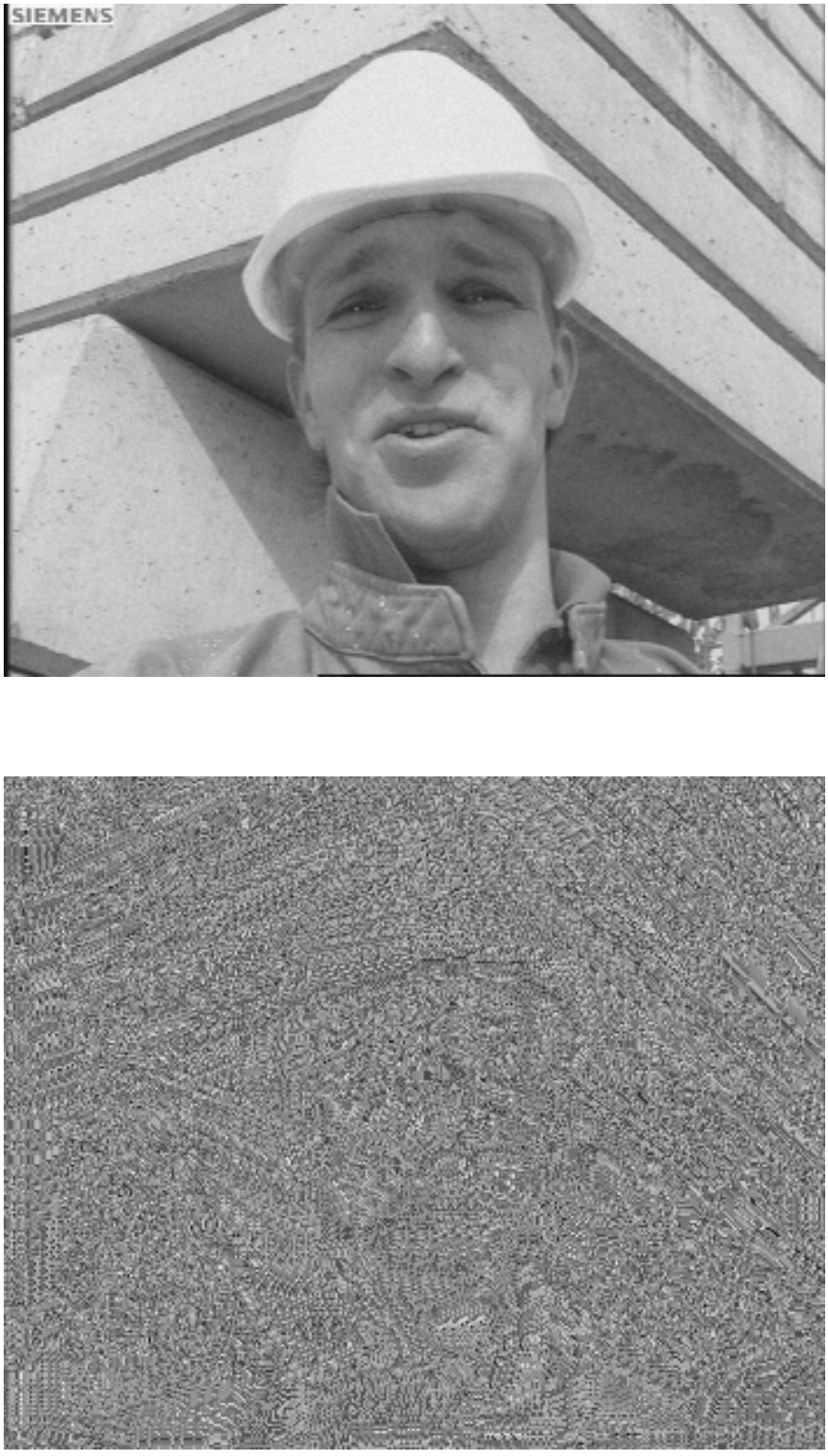


Figure 3.4: A sample frame from the *Foreman* sequence marked using the MPEG-2 watermarking scheme (top) and the corresponding watermark (bottom).

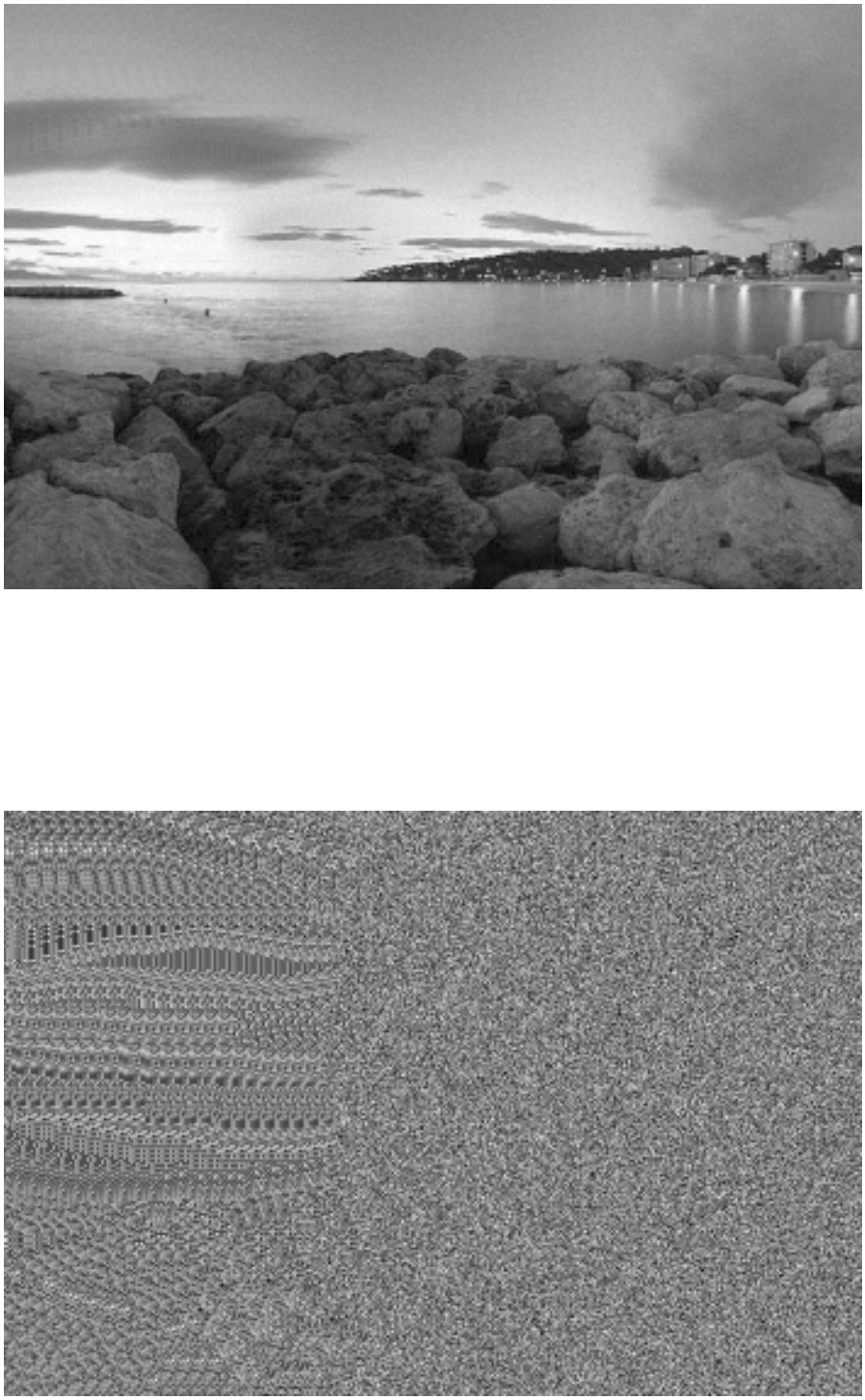


Figure 3.5: A sample frame from the *Antibes* sequence marked using the MPEG-2 watermarking scheme (top) and the corresponding watermark (bottom).

Figure 3.6: A sample frame from the *Coastguard* (top) and *Foreman* (bottom) sequences after the FTF attack.

3.6 Discussion

The danger of the MC-FTF attack can be countered by the MC watermarking schemes in which correlated watermarks are embedded along the motion-trajectories. However, the existing implementations of MC watermarking schemes operate in the uncompressed domain. Such an approach is computationally prohibitive in real-time video watermarking applications since the video sequences are generally stored and transmitted in a compressed format.

This chapter proposed a compressed-domain approach to MC watermarking. Two watermarking schemes have been proposed, one for the MPEG-2 coded sequences and the other for the the sequences coded using the emerging MC-TWT based coding. The experimental results confirm that the proposed schemes offer a good degree of robustness against the MC-FTF attack. Our analysis has shown some desirable properties of MC watermarking like improved robustness to re-encoding and robustness to other known inter-frame collusion attacks. Also, the increase in the bit-rate required to encode a watermarked sequence is less because of the motion coherence of the proposed watermarks.

Chapter 4

An Oracle for Motion-Incoherent Watermarking

Inter-frame collusion attacks have been studied extensively and many watermarking schemes have been proposed in recent years to resist such attacks. However, for a given watermarking scheme, there is no tool available to assess whether the produced watermark is resistant to the inter-frame collusion attacks or not. Today, this assessment relies on a computationally expensive procedure: (1) a sequence is watermarked, (2) the watermarked sequence is subjected to different inter-frame collusion attacks and (3) the detector checks the presence of the watermark in the attacked sequences. Inter-frame collusion resistance can be guaranteed only if the watermark survives all the attacks. Similarly, in the attacker-side, it may be of interest to know whether the inter-frame collusion will be effective on a given watermarked sequence or more importantly, whether an attacked sequence still carries the watermark. This assessment is more challenging since in many applications of watermarking like fingerprinting and copyright protection, the attacker may not have access to the watermark detector. If a tool which can predict the presence of watermark is available to the attacker, the attack efficiency can be improved by changing the attack parameters. For example, the attacker may increase the temporal window width of the FTF attack until the watermark becomes undetectable.

This chapter presents a novel technique to evaluate the inter-frame collusion resistance without going through the expensive attack-detect procedure. As shown in the previous chapter, the motion-coherency in the watermark is a *sufficient* condition to guarantee the resistance to inter-frame collusion. So, assessing the motion-coherency in the watermark is a *sufficiency test* for inter-frame collusion resistance. We propose a simple *oracle* which reports the presence/absence of motion-incoherent watermark in a given sequence. In other words, the oracle reports whether a given sequence carries a watermark

which is *susceptible* to inter-frame collusion or not. The oracle does not make use of any watermark detection algorithm, thereby making it useful to the attacker as well. The proposed oracle exploits the motion-compensated prediction module present in the state-of-the-art video coding techniques. It will be shown that it is possible to extract some statistical features, which can distinguish between the presence of motion-coherent (MC) and that of motion-incoherent (MIC) watermarks, from the motion-compensated prediction error frames. These features are then used to train a pattern classifier which discriminates between the prediction error frames corresponding to a sequence carrying the MC watermark and that carrying the MIC watermark. The proposed system is presented in Section 4.2, after a review of related work in Section 4.1. Experimental results are then reported in Section 4.3 to validate the accuracy of the proposed system.

4.1 Related Work

In a previous work [BK04], a *steganalysis* technique has been introduced to distinguish watermarked video contents from the non-watermarked video material. The underlying idea is to exploit the sensitivity of some watermarking systems against a certain class of collusion attacks. For example, the temporal frame averaging (TFA) can be introduced to detect the presence of uncorrelated watermarks embedded in successive frames.

Many video watermarking algorithms can be reduced to a frame-by-frame additive procedure as written below [DD03a]

$$\mathbf{y}_k = \mathbf{x}_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(0, \sigma_w^2) \tag{4.1.1}$$

where $\mathbf{x}_k$ is the frame at instant k in the host video sequence, $\mathbf{y}_k$ the corresponding watermarked frame and $\mathbf{w}_k$ the watermark signal embedded at instant k, assumed to be normally distributed with zero mean and a variance of σ_w^2. If the watermarked sequence is temporally averaged with an odd window length of $T+1$, the resulting attacked frames $\bar{\mathbf{y}}_k$ are given by

$$\bar{\mathbf{y}}_k = \frac{1}{T+1} \sum_{i \in \mathcal{W}_k} \mathbf{y}_i \tag{4.1.2}$$

where $\mathcal{W}_k = \{k - T/2, k - T/2 + 1, \cdots, k + T/2\}$ is the set of temporal indices in the averaging window. When $\mathbf{y}_k$ is substituted by Equation (4.1.1) in Equation (4.1.2), the following equation is obtained

$$\bar{\mathbf{y}}_k = \frac{1}{T+1} \sum_{i \in \mathcal{W}_k} \mathbf{x}_i + \frac{1}{T+1} \sum_{i \in \mathcal{W}_k} \mathbf{w}_i = \bar{\mathbf{x}}_k + \bar{\mathbf{w}}_k \tag{4.1.3}$$

Assuming that the host frames in the temporal window are perceptually *similar*, the difference frame $\mathbf{z}_k = \mathbf{y}_k - \bar{\mathbf{y}}_k$ between watermarked and attacked contents can be approximated as follows:

$$\mathbf{z}_k = (\mathbf{x}_k - \bar{\mathbf{x}}_k) + (\mathbf{w}_k - \bar{\mathbf{w}}_k) \approx \mathbf{w}_k - \bar{\mathbf{w}}_k \ . \tag{4.1.4}$$

Depending on the sensitivity of considered watermarking algorithm to the TFA, the difference frame $\mathbf{z}_k$ will exhibit different statistics. In particular, if the watermarks embedded in successive frames are uncorrelated, averaging $T+1$ watermarks leads to a normally distributed signal with zero mean and a variance of $\frac{\sigma_w^2}{T+1}$, and the difference frame $\mathbf{z}_k$ can be modelled as

$$\mathbf{z}_k \approx \mathbf{w}_k - \bar{\mathbf{w}}_k \sim \mathcal{N}\left(0, \frac{T}{T+1}\sigma_w^2\right). \tag{4.1.5}$$

Therefore, if the tested sequence is carrying a watermark, $\mathbf{z}_k$ will be Gaussian. On the other hand, if the video is not watermarked, the difference will not be Gaussian. Based on the *Gaussianity* of $\mathbf{z}_k$, one can thus decide whether a watermark is present or not. A classifier can be trained using some features extracted from $\mathbf{z}_k$ to make this decision. In [BK04], the authors proposed to use the *kurtosis*, the *entropy* and the 25^{th} *percentile*.

The kurtosis [RS00] of a random variable X is the forth central moment defined as:

$$\text{Kurtosis} = \frac{1}{\sigma_x^4 N} E(x_i - \mu_x)^4$$

where μ_x and σ_x are respectively the mean and the standard deviation and E is the expectation operator. The kurtosis is the degree of *peakedness* of the corresponding distribution. The kurtosis for a normal distribution is 3 and varies for most of the other distributions.

The entropy is a measure of *randomness* in a given distribution. The entropy estimate is given by

$$\text{Entropy} = -\sum P_X(i) \log P_X(i) \tag{4.1.6}$$

where $P_X(i)$ is an estimate for probability and the summation is over all the estimated probabilities. The entropy estimate from a watermarked sequence is expected to have a higher value as compared to that estimated from the host sequence. Finally, the 25^{th} percentile of a distribution is the value above which 25% of points in the histogram reside.

This steganalysis scheme accurately detects the presence (or absence) of uncorrelated watermarks within static video contents. However, its performance is likely to be severely degraded as soon as some dynamic components, e.g. moving objects and/or camera motion, are present in the video. Indeed, the energy of $(\mathbf{x}_k - \bar{\mathbf{x}}_k)$ in Equation (4.1.4) will no longer be negligible, thus interfering with the

features used for classification. Thus this system is not effective for dynamic video scenes. In order to avoid interferences due to dynamic components, it is necessary to perform motion-compensation before applying the TFA. In other words, the motion-compensated temporal frame averaging (MC-FTA) will be used instead of the conventional TFA operation given in Equation (4.1.2). In a subsequent article [BKZ06], the authors proposed to apply a MPEG-like block-based motion compensation before the TFA. However, the reported experimental results did not clearly exhibit a significant improvement. This can be explained by the fact that they were using videos with little motion activity, thus leading to very little host interference even without motion compensation. The TFA attack module may be replaced with other inter-frame collusion attacks to detect different types of watermarking schemes. For example, if the WER attack is used instead of the TFA, the resulting system should be able to differentiate the sequences carrying SS-1 watermarks. In all the cases, the performance of the system will depend on the efficiency of the attack module.

4.2 Proposed System

The straightforward way to assess the motion-coherency of the watermark in a given sequence is first subjecting the sequence to the MC-FTF attack and then using the watermark detector to check the presence of the watermark in the attacked sequence. However, such a technique may not be useful to the attacker due to the non-availability of the watermark detector. An alternative way is to use some temporal processing which is sensitive to the MIC watermark, such that the change in the processed signal can be used to detect the presence of the MIC watermark. For example, the level of Gaussianity in the difference between the watermarked sequence and its MC-FTF attacked version can be used to detect the presence of the MIC watermark. It should be noted at this point that any such temporal processing should be applied along the motion-trajectories. In other words, the computationally expensive motion estimation for finding the motion trajectories is an indispensable part of any method to assess the motion-coherency of the watermark.

In this Section, first we will show that the *motion-compensated prediction* used in the state-of-the art video coding standards is a temporal process sensitive to the MIC watermarks and it is possible to detect the presence of MIC watermarks from the *motion-compensated prediction error frames*. We then propose a simple *oracle* for MIC watermarks, based on the change in the statistics of the motion-compensated prediction error frames due to the presence of MIC watermarks. The main motivation in using the motion-compensated prediction in the proposed oracle is that the watermarked sequences

are generally stored and transmitted in the compressed format. In such cases, the motion-compensated prediction error frames can be directly obtained by partial decoding of the compressed stream, thereby significantly reducing the computational requirements of the oracle. On the other hand, if the watermarked sequence is in the uncompressed format, motion-estimation and compensation need to be performed to obtain the prediction error frames.

4.2.1 Motion-Compensated Prediction

As mentioned in the previous chapter, in a typical MPEG encoder, there are two types of predicted frames, namely the P-frames and the B-frames. Let the prediction error frame corresponding to a P-frame be denoted by P-PEF and that of a B-frame by B-PEF. The motion-compensated prediction of the frame $\mathbf{x}_{k_2}$ using the reference frame $\mathbf{x}_{k_1}$ is defined as

$$\mathbf{x}_{k_1}^{(k_2)}[\mathbf{n}] \equiv \mathbf{x}_{k_1}\left[\mathcal{M}_{k_1 \to k_2}(\mathbf{n})\right] \approx \mathbf{x}_{k_2}[\mathbf{n}], \quad \mathbf{n} \in \Lambda. \tag{4.2.1}$$

When *sub-pixel* accurate motion compensation is exploited, the point $\mathcal{M}_{t_1 \to t_2}(\mathbf{n})$ might not belong to Λ. In this case, spatial interpolation is performed to obtain the sample values at these sub-pixel locations [ST03]. In the reminder of this analysis, motion-compensation with integer-pixel accuracy is assumed for mathematical tractability. Now a P-PEF is given by

$$\mathbf{P}_{k_2}^{\mathbf{x}} = \mathbf{x}_{k_2} - \mathbf{x}_{k_1}^{(k_2)}, \quad k_1 < k_2 \tag{4.2.2}$$

and a B-PEF by

$$\mathbf{B}_{k_2}^{\mathbf{x}} = \mathbf{x}_{k_2} - \frac{1}{2}\left(\mathbf{x}_{k_1}^{(k_2)} + \mathbf{x}_{k_3}^{(k_2)}\right), \quad k_1 < k_2 < k_3. \tag{4.2.3}$$

In the case of a watermarked video sequence $\{\mathbf{y}_k\}$, obtained using (4.1.1), we get

$$\begin{aligned} \mathbf{y}_{k_1}^{(k_2)}[\mathbf{n}] &= \mathbf{y}_{k_1}\left[\mathcal{M}_{k_1 \to k_2}(\mathbf{n})\right] \\ &= \mathbf{x}_{k_1}\left[\mathcal{M}_{k_1 \to k_2}(\mathbf{n})\right] + \mathbf{w}_{k_1}\left[\mathcal{M}_{k_1 \to k_2}(\mathbf{n})\right] \quad . \end{aligned} \tag{4.2.4}$$

Therefore, we can write

$$\mathbf{y}_{k_1}^{(k_2)} = \mathbf{x}_{k_1}^{(k_2)} + \mathbf{w}_{k_1}^{(k_2)} \quad . \tag{4.2.5}$$

Even though integer-pixel motion-compensation is assumed for obtaining the above equation, it is also valid for sub-pixel motion-compensation with linear interpolation.

A P-PEF of the watermarked sequence can be expressed as

$$\begin{aligned}\mathbf{P}^{\mathbf{y}}_{k_2} &= \mathbf{y}_{k_2} - \mathbf{y}^{(k_2)}_{k_1} \\ &= \mathbf{x}_{k_2} + \mathbf{w}_{k_2} - \left(\mathbf{x}^{(k_2)}_{k_1} + \mathbf{w}^{(k_2)}_{k_1}\right) \\ &= \mathbf{P}^{\mathbf{x}}_{k_2} + \mathbf{P}^{\mathbf{w}}_{k_2} \end{aligned} \tag{4.2.6}$$

where $\mathbf{P}^{\mathbf{x}}_{k_2} = \left(\mathbf{x}_{k_2} - \mathbf{x}^{(k_2)}_{k_1}\right)$ and $\mathbf{P}^{\mathbf{w}}_{k_2} = \left(\mathbf{w}_{k_2} - \mathbf{w}^{(k_2)}_{k_1}\right)$ are the PEFs corresponding to the host signal and the watermark respectively. If the embedded watermark is motion-coherent, the watermark in the current frame can be predicted from the reference frame, i.e. $\mathbf{P}^{\mathbf{w}}_{k_2} \approx \mathbf{0}$. As a result, the PEF of a sequence carrying a motion-coherent watermark will be similar to that of the host sequence. On the contrary, if the embedded watermark is motion incoherent, then $\mathbf{w}_{k_2}$ and $\mathbf{w}^{(k_2)}_{k_1}$ are uncorrelated and the samples of $\mathbf{w}^{(k_2)}_{k_1}$ are independent and identically distributed (iid), following a Gaussian distribution with zero mean and variance $\sigma^2_{\mathbf{w}}$. It should be noted that a point in the reference frame may map to multiple points in the current frame and the iid assumption is violated at such *multiply referred* points. Nevertheless, such points are usually very few in comparison with the total number of points and their impact can be neglected. Since $\mathbf{w}_{k_2}$ and $\mathbf{w}^{(k_2)}_{k_1}$ are iid Gaussian random fields, their difference is also an iid Gaussian random field. Therefore, a P-PEF corresponding to MIC watermark can be modelled as

$$\mathbf{P}^{\mathbf{w}}_{k} \sim \mathcal{N}(0, \sigma^2_{\mathbf{P}^{\mathbf{w}}_{k}}), \qquad \sigma^2_{\mathbf{P}^{\mathbf{w}}_{k}} = 2\sigma^2_{\mathbf{w}}. \tag{4.2.7}$$

Similarly, the B-PEF of a watermarked video sequence is given by

$$\mathbf{B}^{\mathbf{y}}_{k} = \mathbf{B}^{\mathbf{x}}_{k} + \mathbf{B}^{\mathbf{w}}_{k} \tag{4.2.8}$$

and if the embedded watermark is motion incoherent, we can show that

$$\mathbf{B}^{\mathbf{w}}_{k} \sim \mathcal{N}(0, \sigma^2_{\mathbf{B}^{\mathbf{w}}_{k}}), \qquad \sigma^2_{\mathbf{B}^{\mathbf{w}}_{k}} = 1.5\sigma^2_{\mathbf{w}}. \tag{4.2.9}$$

If the motion model well captures the motion in the host video sequence, the energy of $\mathbf{P}^{\mathbf{x}}_{k}$ and $\mathbf{B}^{\mathbf{x}}_{k}$ is negligible and the presence of motion-incoherent watermarks can be detected by simply measuring the level of Gaussianity in $\mathbf{P}^{\mathbf{y}}_{k}$ and $\mathbf{B}^{\mathbf{y}}_{k}$, in a manner similar to the one described in [BK04]. However, the PEF of the host signal depends on many parameters including the accuracy of the motion model, the presence of occluded/uncovered regions, the changes in illumination, *etc*. For instance, the conventional block-based motion-compensation used in the current video coding standards cannot model non-translational motion in the frames accurately, thus resulting in significant energy in the PEF $\mathbf{P}^{\mathbf{x}}_{k}$ and $\mathbf{B}^{\mathbf{x}}_{k}$. In such cases, the performances of an oracle, simply based on the level of Gaussianity in $\mathbf{P}^{\mathbf{y}}_{k}$ and $\mathbf{B}^{\mathbf{y}}_{k}$, are likely to be significantly degraded.

4.2.2 Statistical Modelling of Prediction Error Frames

The motion-compensated prediction error frames generally exhibit highly non-stationary spatial statistics. There may be areas with low energy where the motion-model well captures the motion and areas with high energy where the motion-model is not adequate. In the proposed method, we model each PEF of the host sequence as a *locally iid non-stationary Gaussian* random field which is one of the most widely used statistical models in image processing [MKR99, VDPP01]. This model is characterized by two parameters, the *local mean* and the *local variance*. According to this model, each sample of a PEF follows a Gaussian distribution with specific parameters:

$$\forall \mathbf{n} \in \Lambda, \quad \mathbf{P}_k^{\mathbf{x}}[\mathbf{n}] \sim \mathcal{N}\left(\mu_{\mathbf{P}_k^{\mathbf{x}}}[\mathbf{n}], \sigma^2_{\mathbf{P}_k^{\mathbf{x}}}[\mathbf{n}]\right) \tag{4.2.10}$$

where $\mu_{\mathbf{P}_k^{\mathbf{x}}}[\mathbf{n}]$ and $\sigma^2_{\mathbf{P}_k^{\mathbf{x}}}[\mathbf{n}]$ are respectively the local mean and the local variance.

Let us now analyze how the distribution of the PEFs changes due to the addition of an MIC watermark. As shown in Equations 4.2.6 and 4.2.8, the PEF of a sequence carrying MIC watermark is the sum of two independent random fields : the PEF of the host which is locally iid non-stationary Gaussian and that of the watermark which is iid Gaussian. By using the properties of the sum of independent Gaussian random variables one can show that the distribution the PEFs of a sequence carrying an MIC watermark are also locally iid non-stationary Gaussian, i.e.,

$$\forall \mathbf{n} \in \Lambda, \quad \mathbf{P}_k^{\mathbf{y}}[\mathbf{n}] \sim \mathcal{N}\left(\mu_{\mathbf{P}_k^{\mathbf{y}}}[\mathbf{n}], \sigma^2_{\mathbf{P}_k^{\mathbf{y}}}[\mathbf{n}]\right) \tag{4.2.11}$$

with $\mu_{\mathbf{P}_k^{\mathbf{y}}}[\mathbf{n}] = \mu_{\mathbf{P}_k^{\mathbf{x}}}[\mathbf{n}]$ and $\sigma^2_{\mathbf{P}_k^{\mathbf{y}}}[\mathbf{n}] = \sigma^2_{\mathbf{P}_k^{\mathbf{x}}}[\mathbf{n}] + \sigma^2_{\mathbf{P}_k^{\mathbf{w}}}$. The local mean of the PEFs does not change after the addition of MIC watermark since $\mathbf{P}_k^{\mathbf{w}}$ is assumed to be zero-mean. Similarly in the case of a B-PEF,

$$\forall \mathbf{n} \in \Lambda, \quad \mathbf{B}_k^{\mathbf{y}}[\mathbf{n}] \sim \mathcal{N}\left(\mu_{\mathbf{B}_k^{\mathbf{y}}}[\mathbf{n}], \sigma^2_{\mathbf{B}_k^{\mathbf{y}}}[\mathbf{n}]\right) \tag{4.2.12}$$

where $\mu_{\mathbf{B}_k^{\mathbf{y}}}[\mathbf{n}] = \mu_{\mathbf{B}_k^{\mathbf{x}}}[\mathbf{n}]$ and $\sigma^2_{\mathbf{B}_k^{\mathbf{y}}}[\mathbf{n}] = \sigma^2_{\mathbf{B}_k^{\mathbf{x}}}[\mathbf{n}] + \sigma^2_{\mathbf{B}_k^{\mathbf{w}}}$.

We have seen that the variance parameter of the locally iid Gaussian model of the PEF changes with the addition of MIC watermark. For a model-based description of this change, a suitable model for the local variances is needed. Voloshynovskiy *et.al* [VDPP01] suggested deriving such a model for the local variances of a host image from the *estimated* local variances. We propose to use the estimated local variances to derive similar models for the local variances of the host and the watermarked PEFs.

Since the PEFs are modelled as iid Gaussian within a local window, the *maximum likelihood* (ML) estimator for the local variance of $\mathbf{P}_k^{\mathbf{y}}$ at a point $\mathbf{n}$ in is given as

$$\hat{\sigma}^2_{\mathbf{P}_k^{\mathbf{y}}}[\mathbf{n}] = \frac{1}{N^2 - 1} \sum_{\mathbf{m} \in \mathcal{N}(\mathbf{n})} \left(\mathbf{P}_k^{\mathbf{y}}[\mathbf{m}] - \hat{\mu}_{\mathbf{P}_k^{\mathbf{y}}}[\mathbf{n}]\right)^2 \tag{4.2.13}$$

where $\mathcal{N}(\mathbf{n})$ is the local neighborhood of $N \times N$ points centered at $\mathbf{n}$, and

$$\hat{\mu}_{\mathbf{P}_k^{\mathbf{y}}}[\mathbf{n}] = \frac{1}{N^2} \sum_{\mathbf{m} \in \mathcal{N}(\mathbf{n})} \mathbf{P}_k^{\mathbf{y}}[\mathbf{m}]$$

is the ML estimator for local mean. Substituting the value of $\mathbf{P}_k^{\mathbf{y}}[\mathbf{n}]$ from Equation 4.2.6 and neglecting the cross-terms by using the fact that $\mathbf{P}_k^{\mathbf{x}}[\mathbf{n}]$ and $\mathbf{P}_k^{\mathbf{w}}[\mathbf{n}]$ are independent, Equation 4.2.13 can be approximated as

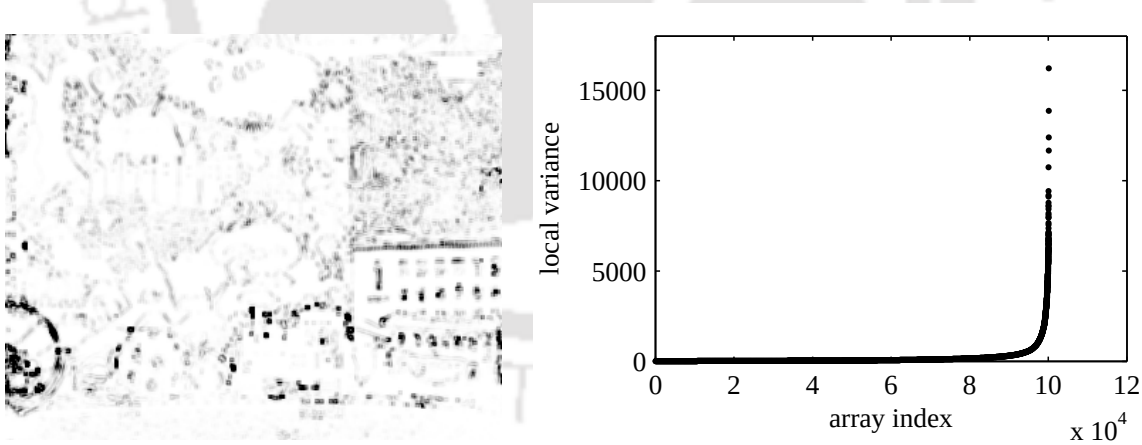
$$\begin{aligned} \hat{\sigma}^2_{\mathbf{P}_k^{\mathbf{y}}}[\mathbf{n}] &\approx \frac{1}{N^2-1} \sum_{\mathbf{m} \in \mathcal{N}(\mathbf{n})} \left(\mathbf{P}_k^{\mathbf{x}}[\mathbf{m}] - \hat{\mu}_{\mathbf{P}_k^{\mathbf{x}}}[\mathbf{n}]\right)^2 + \frac{1}{N^2-1} \sum_{\mathbf{m} \in \mathcal{N}(\mathbf{n})} \left(\mathbf{P}_k^{\mathbf{w}}[\mathbf{m}] - \hat{\mu}_{\mathbf{P}_k^{\mathbf{w}}}[\mathbf{n}]\right)^2 \\ &= \hat{\sigma}^2_{\mathbf{P}_k^{\mathbf{x}}}[\mathbf{n}] + \hat{\sigma}^2_{\mathbf{P}_k^{\mathbf{w}}}[\mathbf{n}] \ . \end{aligned} \tag{4.2.14}$$

Here we assumed that

$$\sum_{\mathbf{m} \in \mathcal{N}(\mathbf{n})} \left(\mathbf{P}_k^{\mathbf{x}}[\mathbf{m}] - \hat{\mu}_{\mathbf{P}_k^{\mathbf{x}}}[\mathbf{m}]\right) \left(\mathbf{P}_k^{\mathbf{w}}[\mathbf{m}] - \hat{\mu}_{\mathbf{P}_k^{\mathbf{w}}}[\mathbf{m}]\right) \simeq 0.$$

Thus the local variance estimated at each point of the MIC watermarked PEF is the sum of the estimated local variances of the corresponding host PEF and the watermark PEF.

Figure 4.1(a) shows the estimated local variance of a P-PEF from the *Mobile* sequence using a 3×3 window. The local variances, estimated from the areas in the PEF corresponding to the occluded/uncovered regions and object boundaries in the current frame, are generally very high compared to those estimated from other areas in the PEF. This is evident from the darker areas in the local variance plot, shown in Figure 4.1(a). The local variance estimates, rearranged to a 1-D array and then sorted in the ascending order, are plotted in Figure 4.1(b). It is clear from this plot that the local



(a) (b)

Figure 4.1: (a) Local variances of a P-PEF from the *Mobile* sequence and (b) the plot of the sorted array of local variances. Darker areas in (a) indicate regions where the estimated variance is high valued.

variance estimated from the above mentioned areas, though a few in numbers, are of extremely high values. These extreme-valued estimates, resulting from the failure of motion-compensated prediction, are considered as *outliers*. This outliers are discarded from the analysis by setting an upper-threshold τ on the local variance estimate.

Motivated by the works [MKRM99, VDPP01] we now derive a statistical model for the ML estimators $\hat{\sigma}^2_{\mathbf{P}^{\mathbf{x}}_k}$. It has been suggested that the histogram of the local variance estimates of the high-pass wavelet bands of an image can be well approximated by using the *exponential*, *Weibull*, *Rice* or *Gamma* distributions. We propose that such approximation is valid for the local variance estimates of the host PEFs. From the above mentioned distributions, we choose the more general *Gamma distribution* to approximate the histogram of $\hat{\sigma}^2_{\mathbf{P}^{\mathbf{x}}_k}$.

The probability density function(PDF) of the two-parameter Gamma distribution [RS00] is given by

$$f(x) = \begin{cases} \frac{b^{-a}}{\Gamma(a)} x^{a-1} \exp[-\frac{x}{b}], & x > 0 \\ 0, & x \leq 0 \end{cases} \tag{4.2.15}$$

where $a > 0$ is the *shape parameter*, $b > 0$ is the *scale parameter* and

$$\Gamma(a) = \int_0^\infty t^{a-1} \exp[-t] dt \tag{4.2.16}$$

is the Gamma function. The shape parameter allows the gamma distribution to take on a wide range of shapes. The scale parameter is a measure of the spread of the distribution. Many distributions like the exponential, Weibull and the chi-squared (χ^2) distributions are the special cases of the Gamma distribution.

The normalized histograms of $\hat{\sigma}^2_{\mathbf{P}^{\mathbf{x}}_k}$, estimated using a 3×3 sliding window and their Gamma approximations for different sequences, are shown in Figure 4.2(a-c). The Gamma approximation parameters are obtained by the maximum likelihood estimation and the statistical tool box of MATLAB. These plots suggest that the Gamma distribution is indeed a reasonable approximation to the distribution of the ML estimator $\hat{\sigma}^2_{\mathbf{P}^{\mathbf{x}}_t}$. In other words, the estimated local variance $\hat{\sigma}^2_{\mathbf{P}^{\mathbf{x}}_t}[\mathbf{n}]$ at each point can be seen as the realization of a Gamma random variable.

Deriving a statistical model for the estimator $\hat{\sigma}^2_{\mathbf{P}^{\mathbf{w}}_k}$ is rather straightforward. Since $\mathbf{P}^{\mathbf{w}}_k$ is an iid Gaussian random field, we can show that the $\hat{\sigma}^2_{\mathbf{P}^{\mathbf{w}}_k}$ will follow a χ^2 distribution with $N^2 - 1$ *degrees of freedom* [RS00]. The PDF of the two-parameter χ^2 distribution is given by

$$f(x) = \begin{cases} \frac{\left(2\sigma^2\right)^{-\frac{n}{2}}}{\Gamma\left(\frac{n}{2}\right)} x^{\frac{n}{2}-1} \exp[-\frac{x}{2\sigma^2}], & x > 0 \\ 0, & x \leq 0 \end{cases} \tag{4.2.17}$$

where $n \in \mathbb{Z}^+$ is the degree of freedom and σ is the scale parameter. On comparing Equations (4.2.15) and (4.2.17), we can see that the χ^2 distribution is a special case of the Gamma distribution with $a = \frac{n}{2}$ and $b = 2\sigma^2$.

The local variance estimator $\hat{\sigma}^2_{\mathbf{P}^{\mathbf{y}}_k}$ can be now seen as the sum of gamma random variables $\hat{\sigma}^2_{\mathbf{P}^{\mathbf{x}}_k}$ and $\hat{\sigma}^2_{\mathbf{P}^{\mathbf{w}}_k}$. Further, it can be assumed that the random variables $\hat{\sigma}^2_{\mathbf{P}^{\mathbf{x}}_k}$ and $\hat{\sigma}^2_{\mathbf{P}^{\mathbf{w}}_k}$ are independent, since they are the functions of independent random variables $\mathbf{P}^{\mathbf{x}}_k[\mathbf{n}]$ and $\mathbf{P}^{\mathbf{w}}_k[\mathbf{n}]$ respectively (Equation (4.2.14)). The sum of the two independent Gamma variables is Gamma distributed only if the scale parameters of the distributions are equal. Nevertheless, the sum of two independent Gamma random variables with unequal scale parameters can still be reasonably *approximated* as a Gamma random variable [Mat82]. Using this approximation, we model the local variance estimators of each watermarked PEF as Gamma distributed. The normalized histograms of the local variances of the PEFs for MIC watermarked sequences and their Gamma approximations are shown in Figure 4.2 (d-f). In summary,

1. The estimated local variances of each host PEF is Gamma distributed.
2. With the addition of the MIC watermark, the estimated local variance of the PEFs are still Gamma distributed, but with different parameters.

This change in the distribution parameters with the addition of MIC watermark is exploited in the proposed oracle.

4.2.3 Estimation of Distribution Parameters

Due to wide range of applications like in the queuing theory and reliability studies, the estimation of the shape and the scale parameters of the Gamma distribution has been an active area of research [BS87, Ada88]. The most popular techniques are the *method of moments* (MM) and the *maximum likelihood* (ML) method. The MM method exploits theoretical formulas of the distribution moments as a function of the scale and shape parameters and equates them with the estimated moments computed from the observations. The resulting system of equation is then solved to estimate the parameters of the distribution [RS00]. On the other hand, ML estimators maximize a likelihood function of the data samples and are obtained by solving a set of likelihood equations. The likelihood equation for the shape parameter of the Gamma distribution does not have a closed-form representation and it is necessary to use numerical techniques such as the Newton-Ralphson method to estimate it [CW69].

Many simple approximations of the ML estimator, which are very close to the actual values, have been proposed. In this work, we use one such method [Dop94], where the ML estimates of the shape

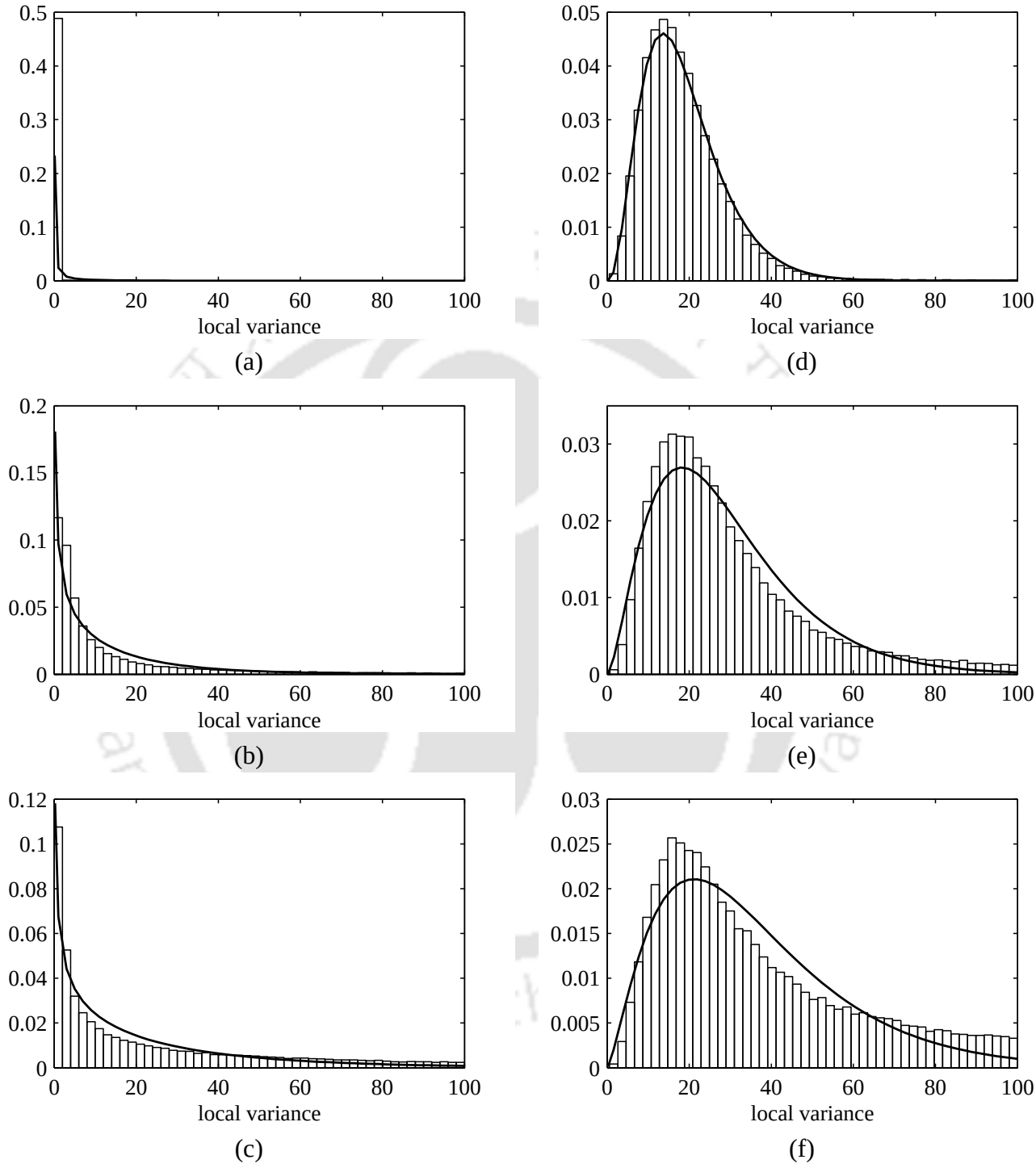


Figure 4.2: Distribution of the local variance of a P-PEF from (a) *Antibes*(host) (b) *Foreman*(host) (c) *Mobile*(host) (d) *Antibes*(MIC watermark) (e) *Foreman(MIC watermark)* and (f) *Mobile*(MIC watermark) sequences

parameter $\hat{a}$ and the scale parameter $\hat{b}$, from N samples $\{x_1, x_2, \cdots, x_N\}$, are approximated as

$$\hat{a} \approx \begin{cases} (\frac{\gamma}{A})^{0.9885} \exp[-0.187(\gamma - A)], & A < \gamma \\ (\frac{\gamma}{A})^{0.8699}, & A \geq \gamma \end{cases} \tag{4.2.18}$$

$$\hat{b} \approx \frac{\hat{\mu}_x}{\hat{a}} \tag{4.2.19}$$

where

$\gamma = 0.577215665$ is the *Euler-Mascheroni constant*

$\hat{\mu}_x = \frac{1}{N}\sum_{i=1}^{N} x_i$ is the sample mean

and

$$A = \ln \frac{1}{N}\sum_{i=1}^{N} x_i - \frac{1}{N}\sum_{i=1}^{N} \ln x_i \ .$$

It has been shown that the relative error in this approximation is not greater than 0.8% .

4.2.4 Oracle Design

In Section 4.2.2, it is shown that there is a statistical difference in the local variance histograms of the PEFs corresponding to a sequence carrying an MC watermark and that carrying an MIC watermark. Based on this difference in the local variance histograms, an oracle for the sequences carrying an MIC watermark is now designed. The first step in the design process is to study how the difference in the histograms are reflected in the distribution parameters of the corresponding Gamma approximations. The study is performed with three test video sequences: *Antibes*, *Foreman* and *Mobile*. The *Antibes* is a synthetic sequence made from a panoramic image with a simple horizontal translation of 2 pixels/frame from left to right. The other two sequences are standard MPEG test sequences with complex motion. The sequences are marked using the SS scheme, which always embeds MIC watermarks. The embedding strength of the watermark is chosen such that a PSNR value of 38 dB is maintained for the watermarked sequences.

First, the PEFs of the uncompressed sequences are analyzed. By using the hierarchical variable size block-matching algorithm [CW99], the P-PEF and B-PEF are computed for both the host video sequences and the watermarked ones. Subsequently, the local variance is estimated in each point by using a 3×3 sliding window. Finally, the distribution parameters of their Gamma approximations are calculated using Equations 4.2.18 and 4.2.19. The estimated scale and shape parameters corresponding to the P-PEFs are plotted in Figure 4.3 for different values of the upper-threshold τ. It is observed that the scale-shape parameters are clustered into two regions: one corresponding to MC watermark

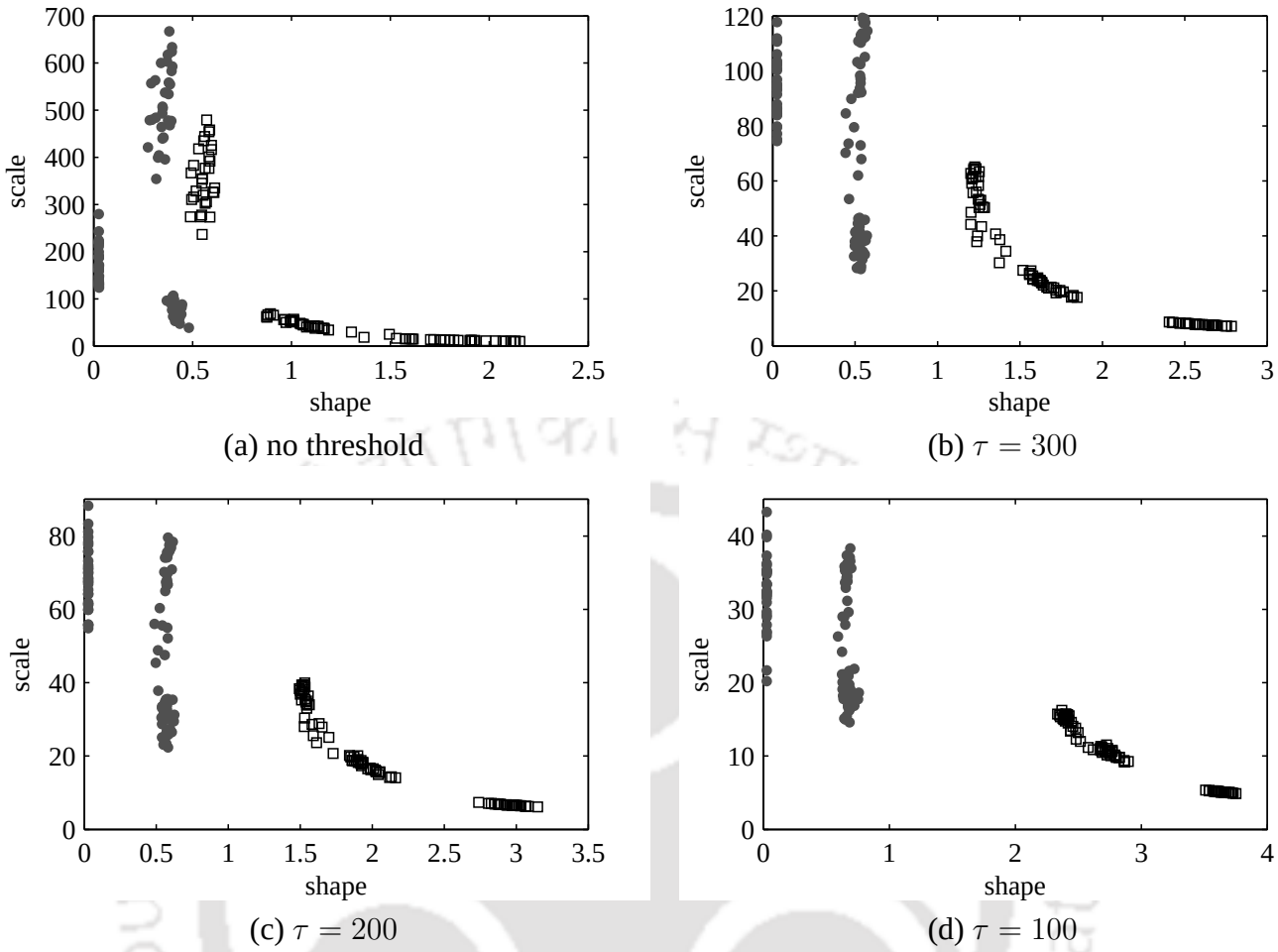
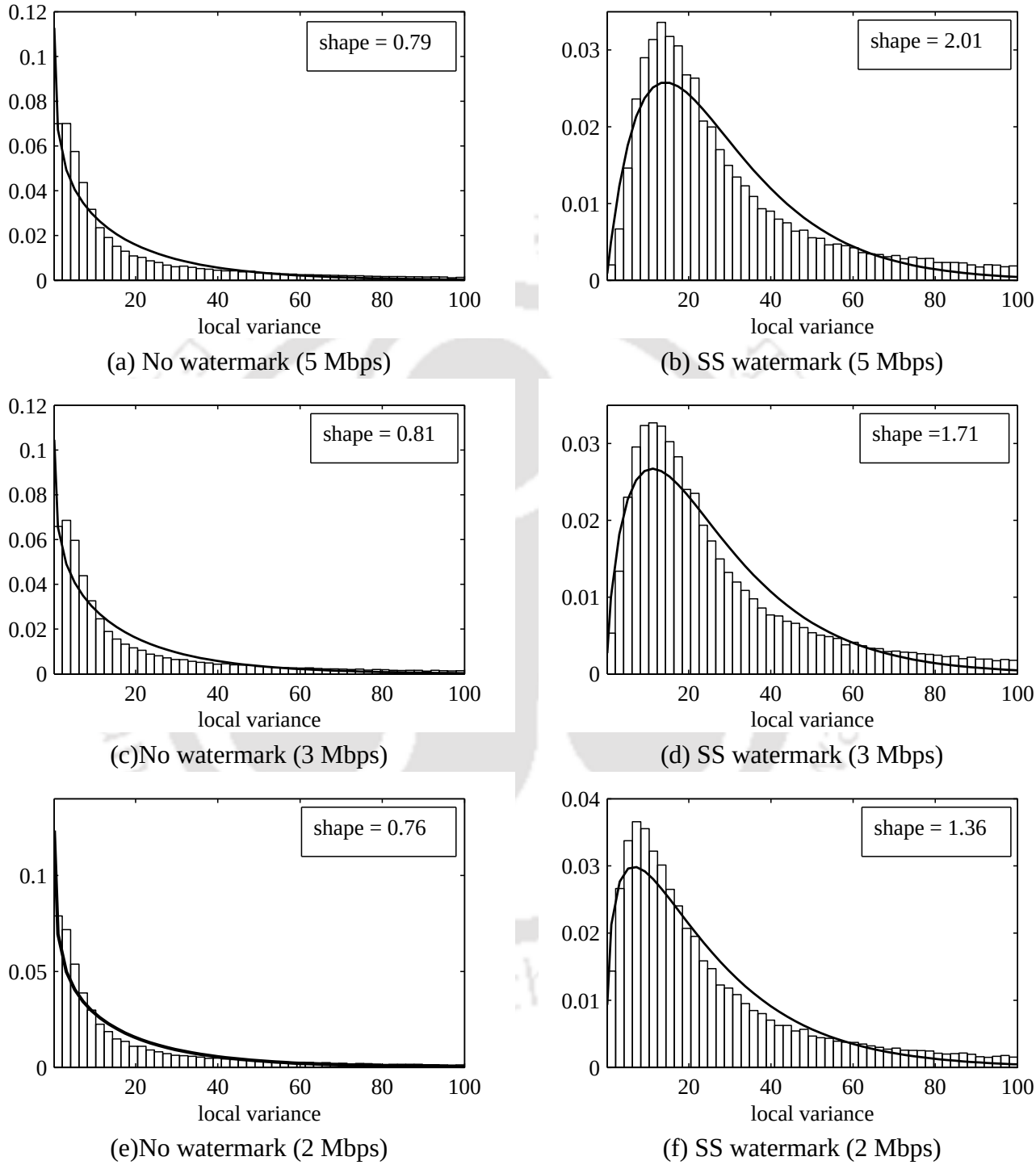


Figure 4.3: Scale-Shape plots corresponding to the P-PEFs from the *Antibes*, *Foreman* and the *Mobile* sequences for different thresholds of the estimated local variance data. Each gray circles (resp. black squares) correspond to one PEF from videos carrying no watermark (resp. a SS watermark).

and the other corresponding to MIC watermark. The clusters are separable even without discarding the outliers in the estimated local variances. However, by setting an upper threshold on the estimated local variance, the clusters become easily separable. The better discriminative power of the shape parameter as compared to the scale parameter is also evident from the plots.

A similar study is conducted for the sequences in the compressed format. The only difference between the PEFs corresponding to the uncompressed and compressed sequences is that the latter is quantized. The quantization step sizes are different for the P-PEFs and the B-PEFs. In a given compressed sequence, the B-PEFs are coarsely quantized compared to the P-PEFs. Due to the non-linear nature of the quantization process, deriving a statistical model for the PEFs of the compressed sequences is not straightforward as we did for the PEFs of uncompressed sequences. Instead, we study whether the statistical model derived for the PEFs of the uncompressed sequences is valid for that of

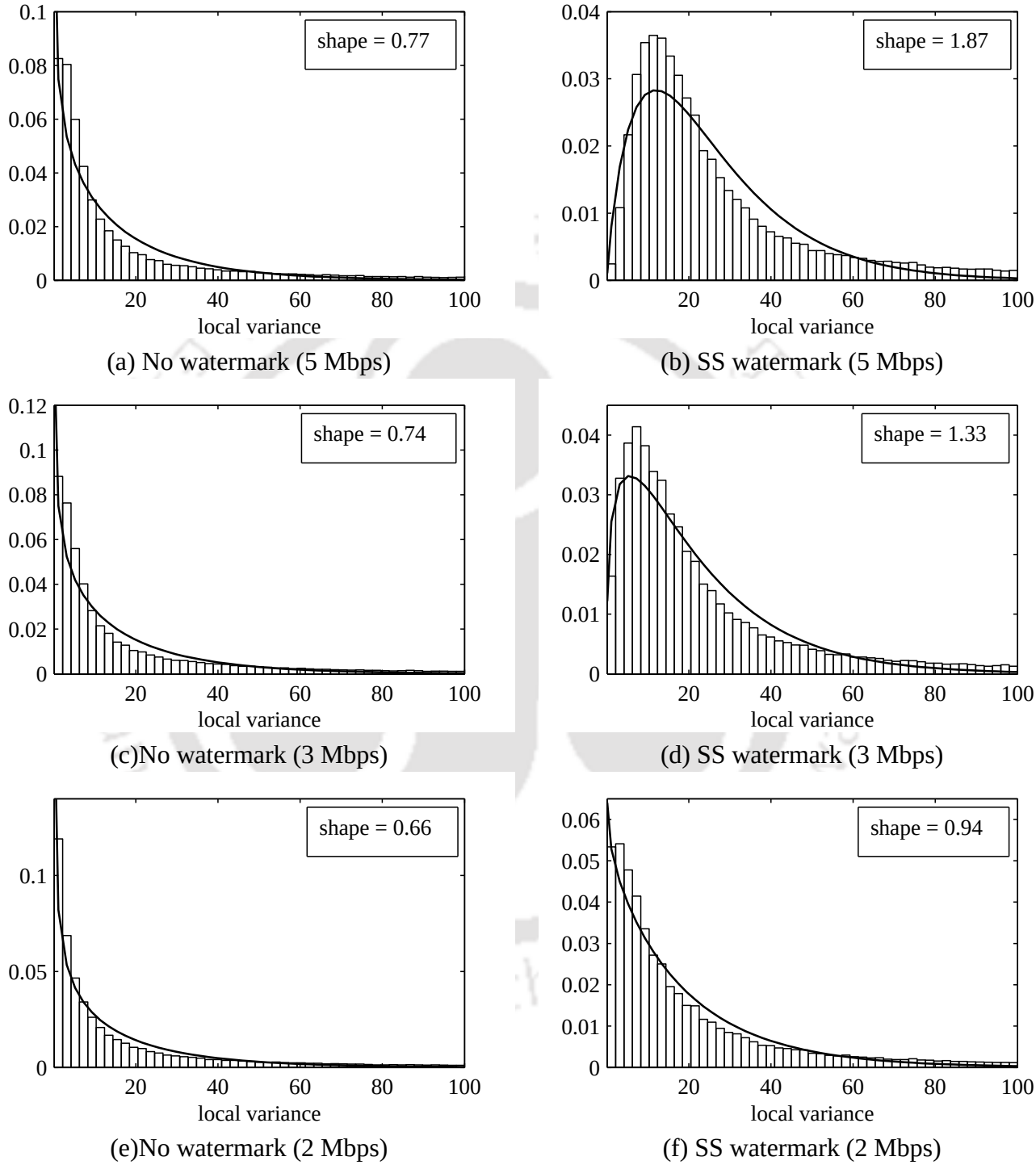


(a) No watermark (5 Mbps) (b) SS watermark (5 Mbps)

(c)No watermark (3 Mbps) (d) SS watermark (3 Mbps)

(e)No watermark (2 Mbps) (f) SS watermark (2 Mbps)

Figure 4.4: Local variance histogram ($\tau = 100$) of a P-PEF from the *Foreman* sequence, coded at different bit-rates and their Gamma approximations.



(a) No watermark (5 Mbps) (b) SS watermark (5 Mbps)

(c)No watermark (3 Mbps) (d) SS watermark (3 Mbps)

(e)No watermark (2 Mbps) (f) SS watermark (2 Mbps)

Figure 4.5: Local variance histogram ($\tau = 100$) of a B-PEF from the *Foreman* sequence, coded at different bit-rates and their Gamma approximations.

the compressed sequences. It should be noted that our objective is to *quantify* the difference in the PEFs corresponding to the sequences carrying an MC watermark and those carrying MIC watermark. This may be achieved with an *approximate* statistical model for the PEFs.

The local variance histograms of the PEFs are studied for the sequences coded using the MPEG-2 standard. The host and the SS watermarked sequences are MPEG-2 coded using the VCDemo software [vcd]. Each sequence is coded at three different bit rates of 5 Mbps, 3 Mbps and 2 Mbps and with a frame rate of 25 frames/second. Note that depending on the bit-rate, a number of 8×8 blocks in the PEFs are quantized to *zero-blocks*. These blocks are excluded from the analysis. Figures 4.4 and 4.5 plot the normalized histograms of the local variance estimates from a P-PEF and B-PEF respectively and the corresponding Gamma approximations for the *Foreman* sequence. From these plots and similar plots for other sequences, it is observed that:

1. The Gamma distribution is a reasonably good approximation to the histograms of the local variance estimated from the PEFs of a compressed sequence.

2. The change in the bit rate has little effect on the histograms corresponding to the host sequence, whereas the histograms corresponding to the sequence carrying an MIC watermark change significantly.

3. For the sequences carrying an MIC watermark, an increase in the compression ratio decreases the shape parameter of the Gamma approximation of the histogram. Due to the coarser quantization as compared to P-PEFs, the decrease in the shape value is more in the case of the B-PEFs.

We now formulate the design of the oracle as a pattern classification problem. The block diagram of the proposed oracle is shown in Figure 4.6. The shape parameter of the Gamma approximation of the local variances estimated from the PEF is used as a feature vector of a pattern classifier which makes the decision that the watermark carried by the test sequence is MC or MIC. The main steps of the oracle can be summarized as:

[1] Compute the PEF with a given motion model

[2] Compute the local variances and discard values exceeding a predecided threshold τ

[3] Compute the shape parameter of the Gamma distribution which best fits the observed local variances

[4] Train a pattern classifier with the estimated shape parameter as the feature vector.

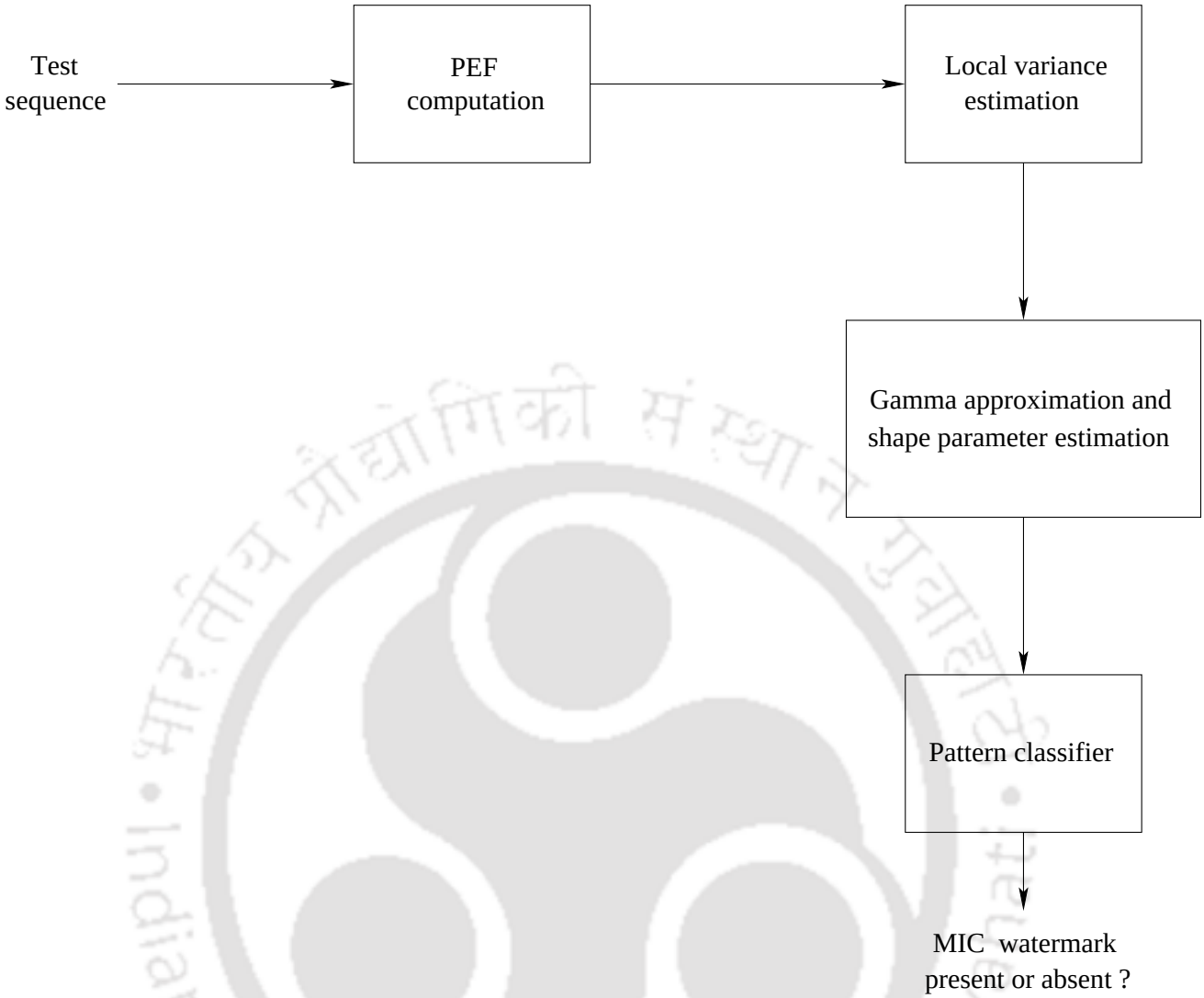


Figure 4.6: Block diagram of the proposed oracle.

4.3 Experimental Results

A number of experiments have been conducted to evaluate the performance of the proposed oracle. A database of 11 grayscale video sequences in raw format has been used during the experiments. Each video consists of 124 frames.The sequences are selected to include a wide spectrum of motion activities. A full description of the sequences is detailed in Table 4.1 with special emphasis on the dynamic components present in the videos such as camera motion and/or moving objects. In contrast with previous studies [BK04, BKZ06], videos with different motion activity have been considered.

Three different watermarking schemes, namely, the MC-TWT based watermarking, SS watermarking and the SS-1 watermarking are considered in the experiments. The MC-TWT based scheme generates MC watermarks and the motion-coherency in the generated watermark was verified in the previous

ID	Sequence name	Frame size	Camera motion	Object motion
1	Container	288×352	N/A	translational (slow)
2	News	288×352	N/A	slow
3	Akiyo	288×352	N/A	slow
4	Mobile	288×352	panning	translational, rotational
5	Coastguard	288×352	panning	translational
6	Tempete	288×352	zooming (fast)	negligible
7	Foreman	288×352	panning, zooming	non-translational (fast)
8	Antibes	240×352	panning	N/A
9	Stefan	240×352	panning (fast)	non-translational (fast)
10	Bike	240×352	panning, tilting (fast)	non-translational (fast)
11	Football	240×352	negligible	non-translational (fast)

Table 4.1: Description of the video sequences used for experiments

chapter in terms of its robustness to the MC-FTF attack. It is implemented using one-level MC-TWT decomposition of the uncompressed sequences, with the Haar wavelet and the HVSBM for estimating the motion-trajectories. The watermark generated by the SS scheme is always MIC, irrespective of the motion content in the video. The SS-1 scheme, on the other hand, generates MC watermarks in the static areas of the scenes and MIC watermarks in the dynamic areas. In all the watermarking schemes, the embedding strength of the watermark is chosen such that the PSNR of watermarked sequence is around 38 dB. In addition to these watermarked sequences, the host sequences for which the response of the oracle is the same as that of sequences carrying an *ideal* MC watermark, are also considered in the experiments.

Although the considered database was originally in raw format, experiments have also been carried out with compressed video sequences. Original and watermarked sequences have been MPEG-2 encoded using the VCDemo software [vcd] at three different bit rates: 5, 3 and 2 Mbps, with a frame rate of 25 frames/sec. The GOP size has been chosen as 12 frames in the IBBPBBPBBPBB format. Therefore, each compressed sequence is made of 11 I-frames, 31 P-frames and 82 B-frames. If the tested video sequence is compressed, the PEFs are retrieved by decoding the corresponding portion of the video stream. For the watermarked sequence in the uncompressed format, the motion vectors are estimated using the HVSBM algorithm with half-pixel accuracy. The intensity values at the sub-pixel locations are estimated using bilinear interpolation. For a fair comparison with the performance on the compressed sequences, the PEFs of the uncompressed sequences are calculated with the same GOP structure as in the compressed ones.

4.3.1 Experiment I: Comparison of Feature Vectors

The first set of experiments is intended to compare the discriminative power of the proposed feature vector (shape parameter of the Gamma approximation of the estimated local variances) and the Guassianity features (kurtosis, entropy and the 25^{th} percentile) proposed in [BKZ06], both extracted from the PEFs. For the comparison, we implement a modified version of the proposed oracle in which the Guassianity features are used instead of the shape parameter. This oracle is referred to as *Oracle-1* and the proposed one as *Oracle-2* hereafter. The host sequences and the watermarked sequences generated by the SS and the MC-TWT based schemes are considered in the experiments. The comparative performance for the uncompressed sequences and the sequences coded at 5 Mbps are reported.

To facilitate a fair comparison, all the steps except the feature extraction are made same in both the oracles. The local variances are estimated from each PEF using a non-overlapping window of size 3×3. The outliers in the estimates are rejected by setting an upper threshold $\tau = 100$, an empirically determined value. Further, the *zero-blocks* in the PEFs obtained from the compressed sequences are not considered in the local variance estimation. The same *pre-processing* steps are applied for computing the Gaussianity features from the PEFs. That is, whenever the local variance estimated from a particular location is greater than τ, the PEFs in the corresponding local window are not considered for computing the Gaussianity features. Likewise, the zero-blocks in the PEFs obtained from the compressed sequences are excluded from the computation of the Gaussianity features.

The oracles employ a *k-nearest neighbor* (kNN) classifier [DHS01] with single nearest-neighbor, the same as the one used in [BK04, BKZ06]. In the kNN classifier, a test object is assigned the label of the majority of the k nearest neighbours according to some distance criterion. For $k = 1$, this is the label of its nearest neighbour in the training set. The feature vectors estimated from the PEFs corresponding to the host and the SS watermarked sequences are used for training the classifiers. Of the available PEFs from these sequences, 40% of the PEFs from each sequence, selected in a random manner, are used for training. Separate classifiers are trained for uncompressed and the MPEG-2 coded sequences. As discussed earlier, the motion-estimation and the quantization steps are different for the P- and the B-PEFs. So, we consider the P-PEFs and B-PEFs separately and separate classifiers are trained for the P-PEFs and the B-PEFs.

First, the host and the SS watermarked sequences are considered. The 60% of the PEFs from these sequences which are not used for training the classifiers are used for testing. For cross-validation [DHS01], the experiment is repeated for 100 different combinations of the training and the test data, chosen in

a random manner. Table 4.2 compares the performance of the oracles for the host and the SS watermarked sequences in the uncompressed format. The values given are the percentage of PEFs which are detected as carrying a MIC watermark. Ideally, the classification results should be close to 0% for the host sequences and close to 100% for SS watermarked sequences whose watermarks are by definition motion-incoherent. As observed from the Table 4.2(a), the Oracle-1 performs well for slow-moving sequences (sequence ID 1 and 2). But, it fails to make correct classification for the P-PEFs corresponding to fast moving sequences. As explained earlier, when the motion estimation fails, the PEFs contain significant contributions from the host frames. In this case, the Oracle-1 which does not consider the host statistics makes wrong classification. On the other hand, the Oracle-2 which considers the host statistics performs perfectly for all the sequences, except for the *Football* sequence. The *Football* sequence contains complex fast and non-translational motion which the block-based motion model cannot capture adequately, resulting in considerably high energy of the PEFs as compared to other sequences. This results in the failure of both the oracles. Even for this sequence, the performance of Oracle-2 is better than that of Oracle-1. Due to better motion-compensation, both the oracles perform better in the case of the B-PEFs compared to the P-PEFs.

The performance of the oracles are now compared for the sequences coded at a bit rate of 5 Mbps. There are two differences between the PEFs obtained from the uncompressed and the compressed sequences. The first difference is in the motion-estimation techniques. For obtaining the PEFs from the uncompressed sequences, we use the HVSBM. On the other hand, the MPEG-2 encoding employs the FSBM with a macro-block size of 16×16 pixels. Due to better motion-estimation with the HVSBM, the PEFs obtained from the uncompressed sequences contain less energy as compared those obtained from the MPEG-2 coded sequences. The other difference is that the PEFs obtained from the coded sequences are quantized. From the results presented in Table 4.3, we can observe a degraded performance of Oracle-1 for the PEFs obtained from the coded sequences in comparison with those obtained from the uncompressed sequences. Note that the classification error is significant in sequences containing fast motion. However the performance of Oracle-2 remains unaffected by poor motion-compensation and the presence of quantization. On comparing the performance of Oracle-1 on the P- and B-PEFs obtained from the coded sequences, it is observed that the oracle performs almost similarly. As compared to the P-PEFs, the B-PEFs undergo better motion-compensation and at the same time are subjected to coarser quantization. The performance improvement due to better motion-compensation may be counterbalanced by the increased quantization step-sizes.

Finally, the oracles are compared in terms of the performance on the sequences other than those

used for training the classifiers. Table 4.4 reports the performance of the oracles for the PEFs obtained from the sequences marked using the MC-TWT domain embedding. Since the embedded watermark is designed to be coherent with motion and thus should not have any MIC component, the classification result should be close to 0%. For the uncompressed sequences, the Oracle-1 classifies the PEFs corresponding to the slow-moving sequences as carrying MC watermarks while many PEFs corresponding to the fast moving sequences are classified as carrying MIC watermarks. The Oracle-2 on the other hand detects the PEFs as carrying MC watermarks, except for a few fast-moving sequences. Like in the previous experiments, both the oracles fail in the case of the *Football* sequence. The performance of the oracles are better in the case B-PEFs compared to the P-PEFs. In the case of the sequences coded at 5 Mbps, the performance of Oracle-1 degrades considerably compared to the case of the uncompressed sequences. A performance degradation is also observed for Oracle-2, especially for the sequences with ID 4, 5, 7 and 10. It should be noted that there is a difference in the motion-estimation techniques used for the watermark embedding (HVSBM) and for computation of the PEFs (FSBM). As a result, when the motion-vectors estimated by using these techniques differ significantly, the PEFs will be detected as carrying MIC watermarks. This difference in the motion-vectors is generally more in the case of sequences with dominant local and non-translational motions. The significant local-motion may be the reason why the Oracle-2 detects a large number of PEFs from sequence with ID 5 as carrying MIC watermarks.

The set of experiments reported above verifies the better discriminative power of the proposed feature vector as compared to the Gaussianity features. In particular, the Gaussianity features fail in presence of poor motion-compensation and quantization. The better discriminative power of the proposed feature vector owes to the incorporation of the host statistics. The performance of the proposed Oracle-2 may be improved by setting larger k in the kNN classifier or using more advanced pattern classifiers like the support vector machine (SVM) [DHS01] instead of the kNN classifier. Such classifiers improve the performance at the cost of increased computational requirements. We have not considered such classifiers since our objective is to develop a simple technique to detect MIC watermarks.

The following experiments evaluate the performance of the proposed Oracle-2 in detail. Since the oracle uses 1-D feature vector, we consider a simple *thresholding classifier* in the remaining experiments. For each PEF, the classifier compares the estimated shape parameter $\hat{a}$ to a threshold τ_c. If $\hat{a} > \tau_c$, then the PEF is detected as carrying an MIC watermark. The threshold τ_c is learnt during the

training process as

$$\tau_c = \frac{\bar{a}_{MC} + \bar{a}_{MIC}}{2} \tag{4.3.1}$$

where $\bar{a}_{MC}$ and $\bar{a}_{MIC}$ are respectively the average of the shape parameters corresponding to the host and the SS watermarked sequences in the training set.

4.3.2 Experiment II: Effect of Compression

This set of experiments studies the performance of the proposed oracle for the sequences coded at different bit-rates. The host, the SS and the MC-TWT domain watermarked sequences, each coded at 3 different bit-rates of 5, 3, and 2 Mbps are considered. The quantization step involved in the coding process can be seen as a watermark removal operation. So, for analyzing the oracle performance on the coded sequences, we need to consider how much of the watermark survives quantization. For this, we study the NC performance of the decoded sequences. The NC performance of the SS and the MC-TWT domain watermarked sequences, coded at different bit rates, are presented in Table 4.5. The better coding performance of the MC-TWT domain watermarking, which generates MC watermarks, is evident from the high NC values. Also note that due to coarser quantization, more watermark components are removed from the B-frames as compared to the P-frames.

Tables 4.6-4.8 present the performance of the oracle on the coded sequences, along with the performance on the corresponding uncompressed sequences for comparison. Ten percentage of the PEFs from the host and the SS watermarked sequences is used for training the classifiers. The classifiers for the coded sequences are trained using the sequences coded at the highest bit-rate (5 Mbps). Like in the previous experiments, the experiments are repeated for 100 different training sets chosen in a random manner and the average performance is reported.

As reported in Table 4.6, the oracle correctly detects the host sequences, except the *Football* sequence, as not carrying any MIC watermark. For the *Football* sequence, the oracle performs better in the case of the uncompressed sequences due to the better motion-compensation using the HVSBM. Also, the number of PEFs detected by the oracle as carrying MIC watermarks decreases with reducing the bit-rate. It is also observed that the performance of the oracle is better on the B-BEFs as compared to the P-PEFs. This may be attributed to the better motion-compensation in the B-PEFs.

The performance of the oracle on SS watermarked sequences is presented in Table 4.7. As expected, the number of PEFs detected by the oracle as carrying MIC watermark decreases with the reduction in the bit-rate. This is in accordance with the corresponding NC values presented in Table 4.5. For a given

bit-rate, the oracle detects more number of P-PEFs as compared to the B-PEFs. The sequences with IDs 4, 6, 7 and 9 contain more complex motion than other sequences, and include considerable host contribution to the corresponding PEFs. This may be the reason behind the low detection rate for these sequences, especially for the P-PEFs at 2 Mbps and the B-PEFs at 3 Mbps. The oracle completely fails to detect the B-PEFs from the sequences coded at 2 Mbps. Note that the effect of quantization at this bit-rate is such that even the watermark detector fails to detect it. The NC values give an idea about the oracle performance. However, no direct relationship can be established between the oracle performance and the NC values. For example, the oracle detects 76.3% of the P-PEFs from the sequence with ID 5 coded at 2 Mbps, where the NC value is 0.58. However, it fails to detect any P-PEF from the sequence with ID 9 coded at the same bit-rate as carrying MIC watermark though the corresponding NC value is 0.59. It should be noted that the watermark detector is non-blind and has access to the host sequence and the original watermark, whereas the oracle is based only on the statistics of the host and the watermark.

Table 4.8 reports the performance of the oracle on the sequences marked using the MC-TWT domain watermarks. As mentioned earlier, there is a difference in the motion estimation technique used for watermark embedding and MPEG-2 encoding processes. This results in the detection of some PEFs from the coded sequences as carrying the MIC watermark. For the sequences coded at different bit rates, the oracle performs in a non-uniform manner. For some sequences, the number of PEFs detected as carrying MIC watermarks decreases with a reduction in the bit-rate while the number increases with the reduction in the bit-rate for other sequences. This can be explained with the motion-estimation and the quantization processes in MPEG-2 encoding. In the encoding process, the reference frame used for the motion estimation is obtained by replicating the decoding process. That is, the motion estimation is done using the quantized reference frame. It is likely that the motion vectors used for watermark embedding differ from those estimated during the encoding process. As the quantization step size increases, these differences are also likely to increase, thereby introducing more watermark components in the PEFs. So, the oracle performance depends on how much watermark components are introduced into the PEFs due to the mismatch between the estimated motion vectors during the watermark embedding and MPEG-2 encoding, and also on how much of the introduced watermark in the PEFs are removed by the quantization process. The combined effect of the motion vector mismatch and the quantization may be the reason behind the non-uniform performance of oracle for the coded sequences at different bit rates.

4.3.3 Experiment III: Hybrid MC watermarking

The final set of experiments evaluates the performance of the oracle on the SS-1 watermarked sequences. Since the SS-1 scheme embeds MC watermarks in the static areas of the frames and MIC watermarks in the dynamic areas, it can be seen as a *hybrid* MC watermarking scheme. The oracle, which checks *global* motion-coherency in the watermark, may fail for such sequences. A practical way to handle this situation is to separate the static and dynamic areas in the PEFs such that the oracle should be able to detect MIC watermarks from the static and dynamic parts. The motion vectors corresponding to the PEFs may be used to find the static and dynamic areas in the PEFs. In this way, the separation of the static and dynamic areas requires no additional computations.

Table 4.9 shows the performance of this modified oracle on the sequences in the uncompressed format (P-PEFs). The training process in the classifier is same as that used in the previous set of experiments. That is, the host and the SS watermarked sequences are used for training. Note that the separation of the static and dynamic areas is not required during the training process since the SS schemes embeds MIC watermarks in both the static and dynamic areas in the frames. In some sequences, the static areas are very small compared to the total area of the PEF, and this may result in erroneous performance of the oracle due to insufficient samples for the computation of the shape parameter. So, if the static area is less than 5% of the total area of the PEF, it is not considered for the classification and is denoted by 'N/A' in the Tables. The classification results clearly indicate that the type of area has no impact except for the SS-1 watermarking algorithm. In this case, the oracle accurately detects that the embedded watermark is motion coherent in static areas and motion incoherent in dynamic ones. As a result, relying on such content-based classification, one can classify most of the watermarking schemes as suggested in Table 4.10. Such a classification scheme is advantageous for an attacker in choosing the appropriate attack. For example, if the watermarking scheme is detected as SS, there is no point in attacking the sequence with WER attack. It should be noted that, to the best knowledge of the author, no hybrid watermarking system which would combine a motion coherent-component in dynamic areas and a motion-incoherent component in static areas has been proposed yet.

ID	NO watermark		SS watermark	
	Oracle-1	Oracle-2	Oracle-1	Oracle-2
1	0.8	0	100	100
2	0	0	100	100
3	1	0	80.1	100
4	0.5	0	98.5	100
5	24.7	0	100	100
6	1.8	0	93.4	100
7	0	0	89.4	100
8	0	0	96.3	100
9	0.5	0	94.5	100
10	13.8	0	90.1	100
11	70.8	57.6	89.3	100
Avg.	10.4	5.2	93.8	100

(a) P-PEF

ID	NO watermark		SS watermark	
	Oracle-1	Oracle-2	Oracle-1	Oracle-2
1	0	0	100	100
2	0	0	100	100
3	0	0	96.5	100
4	0	0	100	100
5	5.9	0	100	100
6	0	0	96.2	100
7	0	0	100	100
8	0	0	100	100
9	0	0	100	100
10	0	0	91.8	100
11	70.1	22	100	100
Avg.	6.9	2	98.6	100

(b) B-PEF

Table 4.2: Comparative performance of Oracle-1 and Oracle-2 on the host and the SS watermarked sequences in the uncompressed format in terms of the percentage of frames detected as carrying MIC watermarks.

ID	NO watermark		SS watermark	
	Oracle-1	Oracle-2	Oracle-1	Oracle-2
1	4.5	0	100	100
2	0.1	0	94.8	100
3	0.7	0	56.9	100
4	32.3	0	83.6	100
5	50.7	0	90.9	100
6	39.4	0	90.8	100
7	23.8	0	84	100
8	1.1	0	69.9	100
9	9.1	0	62.9	100
10	25.6	0	83.1	100
11	86.8	83.3	98.3	100
Avg.	24.9	7.6	83.2	100

(a) P-PEF

ID	NO watermark		SS watermark	
	Oracle-1	Oracle-2	Oracle-1	Oracle-2
1	4.3	0	100	100
2	0	0	97.4	100
3	2.6	0	57.1	100
4	33.7	0	79.3	100
5	51.3	0	97.2	100
6	42.4	0	92.6	100
7	20.1	0	82.2	100
8	1.2	0	78.9	100
9	15.8	0	77.4	100
10	9.6	0	69.9	100
11	87	71.4	98.3	100
Avg.	24.4	6.5	84.6	100

(b) B-PEF

Table 4.3: Comparative performance of Oracle-1 and Oracle-2 on the host and the SS watermarked sequences coded at 5Mbps in terms of the percentage of frames detected as carrying MIC watermarks.

ID	P-PEF		B-PEF	
	Oracle-1	Oracle-2	Oracle-1	Oracle-2
1	0.7	0	0	0
2	0.7	0	0	0
3	0.8	0	0	0
4	42.9	0	10.4	0
5	47.6	6.7	12.5	0.8
6	26.4	6.5	5.3	0
7	9.5	3.2	0	0
8	3	0	0	0
9	8.6	5.8	1.6	0
10	47.3	7	0	0
11	73.6	85.3	79.3	61.8
Avg.	23.7	10.4	9.9	5.7

(a) uncompressed

ID	P-PEF		B-PEF	
	Oracle-1	Oracle-2	Oracle-1	Oracle-2
1	65.6	0.3	35.9	13.2
2	0.5	0	0	5
3	0.8	0	0.7	1.3
4	54.8	29	51.2	25.4
5	63.9	75.7	57.5	50.4
6	66.7	9	66.8	0
7	65.2	31.3	43.2	17.8
8	0.4	0	0.2	0
9	34.6	2	42.4	0
10	73.5	26.8	20	18.8
11	90.5	100	88.6	87.5
Avg.	47	24.9	37	19.9

(b) coded at 5 Mbps

Table 4.4: Comparative performance of Oracle-1 and Oracle-2 on the MC-TWT domain watermarked sequences in terms of the percentage of frames detected as carrying MIC watermarks.

ID	P-frames			B-frames		
	5 Mbps	3 Mbps	2 Mbps	5 Mbps	3 Mbps	2 Mbps
1	0.98	0.81	0.64	0.83	0.55	0.34
2	1.00	0.84	0.64	0.86	0.58	0.37
3	0.98	0.81	0.63	0.82	0.54	0.31
4	0.9	0.71	0.53	0.68	0.38	0.23
5	0.92	0.74	0.58	0.76	0.46	0.26
6	0.89	0.71	0.54	0.72	0.42	0.25
7	0.96	0.76	0.62	0.80	0.50	0.28
8	1.00	0.92	0.76	0.92	0.66	0.41
9	0.95	0.76	0.59	0.80	0.51	0.29
10	1.00	0.83	0.66	0.89	0.59	0.35
11	0.92	0.74	0.56	0.77	0.46	0.29

(a) SS watermarking

ID	P-frames			B-frames		
	5 Mbps	3 Mbps	2 Mbps	5 Mbps	3 Mbps	2 Mbps
1	1.00	0.98	0.93	1.00	0.97	0.88
2	1.00	1.00	0.98	1.00	1.00	0.98
3	1.00	0.99	0.95	1.00	0.98	0.92
4	0.96	0.86	0.74	0.90	0.77	0.65
5	0.98	0.87	0.76	0.93	0.80	0.67
6	0.97	0.85	0.73	0.89	0.73	0.60
7	0.99	0.89	0.75	0.92	0.75	0.59
8	1.00	1.00	1.00	1.00	1.00	1.00
9	0.98	0.88	0.75	0.93	0.78	0.64
10	1.00	0.94	0.82	0.98	0.83	0.67
11	0.96	0.84	0.71	0.89	0.72	0.58

(b) MC-TWT watermarking

Table 4.5: NC performance of the watermarked sequences after coding at different bit rates.

ID	Uncompressed	5 Mbps	3 Mbps	2 Mbps
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0.5
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0
11	19.4	82.8	75.4	57.8
Avg.	1.8	7.5	6.9	5.3

(a) P-PEF

ID	Uncompressed	5 Mbps	3 Mbps	2 Mbps
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0
11	3.5	62.1	24.1	2.5
Avg.	0.3	5.6	2.2	0.2

(b) B-PEF

Table 4.6: Performance of the proposed oracle on the host sequences in the uncompressed and compressed formats in terms of the percentage of frames detected as carrying MIC watermarks.

ID	Uncompressed	5 Mbps	3 Mbps	2 Mbps
1	100	100	100	93.1
2	100	100	100	100
3	100	100	100	89
4	100	100	89.8	23.3
5	100	100	100	95.1
6	100	100	77.2	1.3
7	100	100	100	31.3
8	100	100	100	76.3
9	100	100	66.4	0
10	100	100	100	92.3
11	100	100	99	77.4
Avg.	100	100	93.9	61.7

(a) P-PEF

ID	Uncompressed	5 Mbps	3 Mbps	2 Mbps
1	100	100	98.3	0
2	100	100	100	0
3	100	100	92.3	0
4	100	100	37.1	1.3
5	100	100	71.8	5.5
6	100	97.8	0	0
7	100	100	12.4	0
8	100	100	99.5	14.2
9	100	99.8	8.8	0
10	100	100	72.3	0.1
11	100	100	58.8	10.1
Avg.	100	99.8	59.2	2.8

(b) B-PEF

Table 4.7: Performance of the proposed oracle on the SS watermarked sequences in the uncompressed and compressed formats in terms of the percentage of frames detected as carrying MIC watermarks.

ID	Uncompressed	5 Mbps	3 Mbps	2 Mbps
1	0	0	0.1	15.8
2	0	0	0	0
3	0	0	0	0
4	0	29.3	10.1	0
5	0	73.5	76	47.5
6	0	7.8	0	0
7	0	31	34.1	0
8	0	0	0	0
9	0	1.7	0	0
10	0	27	27.6	15.7
11	67.7	100	86.2	70.3
Avg.	6.2	24.6	21.3	13.6

(a) P-PEF

ID	Uncompressed	5 Mbps	3 Mbps	2 Mbps
1	0	1.3	7.4	8.2
2	0	5	1.1	0
3	0	1.3	0	0
4	0	16.3	2	0
5	0	41	19	0
6	0	0	0	0
7	0	11.1	1.7	0
8	0	0	0	0
9	0	0	0	0
10	0	13.6	0.2	0
11	16.8	87.5	43.7	7.5
Avg.	1.5	16.1	6.8	1.4

(b) B-PEF

Table 4.8: Performance of the proposed oracle on the MC-TWT domain watermarked sequences in the uncompressed and compressed formats in terms of the percentage of frames detected as carrying MIC watermarks.

ID	NO watermark	MC-TWT watermark	SS-1 watermark	SS watermark
1	0	0	0	100
2	0	0	0	100
3	0	0	0	100
4	0	0	0	100
5	0	0	0	100
6	N/A	N/A	0	N/A
7	0	0	0	100
8	N/A	N/A	0	N/A
9	0	0	11	100
10	N/A	N/A	0	N/A
11	36.1	78.1	66	100
Avg.	4.5	9.8	7	100

(a) static areas

ID	NO watermark	MC-TWT watermark	SS-1 watermark	SS watermark
1	0	0	100	100
2	0	0	100	100
3	0	0	100	100
4	0	0	100	100
5	0	9	100	100
6	0	0	100	100
7	0	3.1	100	100
8	0	0	100	100
9	0	0	100	100
10	0	0	100	100
11	22.5	75.5	100	100
Avg.	2	8	100	100

(b) dynamic areas

Table 4.9: Performance of the oracle on the static and dynamic areas from sequences in the uncompressed format in terms of the percentage of frames detected as carrying MIC watermarks

		Static areas	
		MC	*MIC*
Dynamic areas	*MC*	MC-TWT, Host	N/A
	MIC	SS-1	SS

Table 4.10: Classification of watermarking algorithms depending on the oracle response in static and dynamic areas.

4.4 Discussion

The motion-coherency in the watermark has been identified recently as a desirable property to counter the MC-FTF attack. Although several MC watermarking schemes have been recently proposed, there is no simple tool to assess motion coherency in the watermark generated by a given watermarking system. Such a tool will be useful to the watermark designers to benchmark their watermarking system and attackers to fine-tune their attack.

In this chapter, a simple oracle has been designed to assess the motion-coherency in the watermark. The oracle accurately reports whether a given watermarked sequence contains any motion-incoherent component or not. The basic idea behind the oracle is to exploit the statistical changes in the PEFs due to the addition of an MIC watermark. It has been shown that the PDF of the local variance estimated from a PEF can be reasonably approximated with a 2-parameter Gamma distribution. The addition of MIC watermarks changes the shape parameter of the Gamma distribution. Using the shape parameter of the Gamma approximation as the feature vector, a pattern classifier is trained to make the decision on the motion-coherency in the watermark. The reported experimental results on a number of sequences, both in the compressed and uncompressed format, clearly demonstrate the accuracy of the proposed oracle. On important advantage of the oracle is that the PEFs can be directly obtained from a compressed stream.

Chapter 5

Conclusions

Security issues in watermarking have become an active concern of the watermarking researchers. In the context of video watermarking, an attack on watermark security that has received an increasing interest in the recent years is the inter-frame collusion attack. One such effective inter-frame collusion attack is the *frame temporal filtering* (FTF) that removes uncorrelated watermarks embedded in successive frames of a sequence. The thesis investigated how the motion-information in the video frames can be exploited by the attackers to design more effective FTF attacks and the methods to counter such attacks. The main contributions of the thesis are summarized in Section 5.1 and a few tracks for future research are outlined in Section 5.2.

5.1 Summary of Contributions

The thesis addressed three problems related to the inter-frame collusion attack:

- Development of an efficient motion-compensated frame temporal filtering (MC-FTF) attack
- Development of motion-coherent watermarking schemes for compressed video and
- Development of an oracle for assessing the presence of motion-incoherent watermark in video.

The first problem addressed by the thesis is how to improve the performance of the inter-frame collusion attacks by exploiting the motion-information in the video frames. We proposed a new MC-FTF attack based on the *motion-compensated redundant temporal wavelet transform* (MC-RTWT). In this attack, the watermarked frames are temporally filtered along the motion trajectories using the MC-RTWT to remove uncorrelated watermarks. The temporal low-pass frames of the MC-RTWT decomposition

of video constitute the attacked video. The MC-RTWT is implemented through block-based motion compensation and a lifting -based implementation of the wavelet transform. It has been shown that the perceptual quality of the attacked sequence can be improved by making the update lifting step content adaptive. The experimental results presented in Chapter 2 demonstrated the effectiveness of the proposed attack. The main advantages of the proposed MC-FTF attack over the existing FTFR attack are:

1. It is computationally simple due to the use of the block-based motion estimation and the lifting-based implementation of the RTWT; and
2. Unlike the FTFR attack, it removes the watermarks from both the foreground and background regions.

Detailed analysis on the impact of different watermarking schemes on the estimated motion vectors are also presented. The analysis shows that the SS watermarking scheme does not have any impact on the estimation of motion vectors. But the SS-1 watermarks biases the motion-estimation and in this case, better attack performance can be achieved if the motion-vectors estimated from the host sequence are available to the attacker. Another observation is that, if a watermarked video sequence is coded with an MC-TWT based coding technique, the embedded watermark will undergo unintentional MC-FTF attack.

A *motion-coherent* (MC) watermark is a counter-measure to the MC-FTF attack. Chapter 3 investigated the development of computationally-efficient MC watermarking schemes for compressed videos. The chapter first analyzed how the existing video watermarking schemes perform against the inter-frame collusion attacks. We proposed two MC watermarking schemes, one for the MPEG-2 based coding and the other for the emerging MC-TWT based coding.

One of the problems associated with the existing MPEG-2 based watermarking schemes is the problem of watermark-drifting. Because of the closed-loop prediction structure, the watermark added to the I and P frames propagate to other inter-coded frames P and B frames during decompression, thus affecting the video quality. To cancel the drift effect, a drift-compensation signal is added to each inter-coded frame. We analyzed the drift signal in detail and showed that instead of cancelling the drift signal, its proper use may generate MC watermarks. In the proposed MPEG-2 watermarking scheme, MC watermarks are added only to the I-frames of each GOP. During decompression, the watermarks in the I-frames propagate to the inter-coded frames according to the motion in the sequence and generate MC watermarks for these frames. The experimental results confirmed the effectiveness of

the watermarking scheme against the MCFTF and other known inter-frame collusion attacks like the FTF and the watermark estimation remodulation (WER) attack.

Because of the inherent scalability of the 3-D wavelet coding, the MC-TWT based video coding has emerged as a strong alternative to the hybrid video-coding schemes. The proposed MC-TWT based watermarking scheme is developed for the sequences coded using the MC-TWT video coding. In this watermarking scheme, the coded stream is first partially decoded to obtain the temporal low-pass frames and then the watermark is added to each of the temporal low-pass frames. We have shown that the watermark embedded in the temporal low-pass frames generates MC watermarks for the entire sequence during decompression. The reported experimental results verified the robustness of the proposed MC-TWT based watermarking against the MC-FTF and other inter-frame collusion attacks.

The proposed watermarking schemes are computationally efficient and conceptually simple. Detailed analysis presented in the Chapter 3 has shown that the proposed schemes and the MC watermarking in general, offer some desirable properties like less impact of bit-rate control[1] and improved robustness to re-encoding. We have also shown that the motion-coherency in the watermark is a sufficient condition to guarantee robustness against all the known inter-frame collusion attacks.

The final contribution of the thesis is the design of an oracle which detects whether a given video sequence contains any motion-incoherent watermark or not. The oracle is based on the statistical analysis of the motion-compensated prediction error frames (PEF). It was shown that the addition of MC watermarks does not change the statistical properties host PEFs. The addition of MIC watermarks, on the other hand, changes the statistical properties of the host PEFs. A model-based approach is followed to characterize this change in statistics. We have shown that the PEFs can be modelled as a locally non-stationary Gaussian random field which is parameterized by the local mean and the local variance. The presence of motion-coherent watermarks changes the local-variance whereas the local-mean remains unaffected. This change in the local-variance is captured by the corresponding change in the histogram of the estimated local-variance. It was shown that the histogram of the estimated local-variance from a PEF can be reasonably approximated with a 2-parameter Gamma distribution. With the shape parameter of the Gamma distribution as the feature vector, a pattern classifier is trained. The classifier discriminates between sequences carrying MC and MIC watermarks. Experimental studies conducted on both the uncompressed and compressed sequences demonstrated that the oracle detects the presence of motion-incoherent watermark with a good accuracy. We have shown that running the oracle in

[1] In a recent work [HMT07], it has been reported that about 20% reduction in the bit-rate can be achieved with motion-coherent watermarks.

parallel for static and dynamic areas in the frames permits to deal with hybrid motion-coherent watermarking schemes such as the SS-1 watermarking. One added advantage of the oracle is that the PEFs can be directly obtained from the hybrid-coded video data.

5.2 Tracks for Future Work

Several issues related to motion-coherent video watermarking still remain open. A few tracks for future research out of the present work are outlined below.

- **Better motion model:** The performance of the proposed MC-FTF attack, MC watermarking schemes and the oracle to detect the presence MIC watermarks depends on the accuracy of the motion model. Better performance may be achieved by replacing the block-based motion estimation by advanced motion estimation techniques.

- **Blind-detection:** The proposed watermarking schemes use non-blind detection which requires both the host sequence and the original watermark. This is indeed a strong constraint on their practical applicability. Therefore, further investigation is needed to develop blind detectors for the proposed watermarking schemes.

- **Temporal synchronization:** Video watermarking schemes which embed different watermarks in different frames are vulnerable to temporal desynchronizing operations like frame dropping, frame insertion and frame-rate changes. These operations hardly affect the quality of the video but have adverse impact on the detectability of the watermark, particularly in the blind detectors. So, the temporal re-synchronization during the detection of MC watermarks is an important area to be investigated.

- **Collusion-resistant fingerprinting:** In the video fingerprinting applications, the watermarks need to survive both the inter-frame and inter-video collusion attacks. However, the watermarking community has been investigating these two problems independently. Even though significant theoretical advancement has been achieved in the development of collusion-resistant fingerprints [BS98, Tar03, WTWL04], little attention has been paid so far to its practical implementations [HW06, VHS+07]. In this regard, it would be interesting to investigate how the collusion-resistant fingerprints can be embedded in the video maintaining the motion-coherency in the watermark.

- **Signal-coherent watermarking:** The block replacement attack (BRA), though computationally demanding, has been shown recently as an effective way to sever detector performance in still image watermarking [DDK06]. The basic idea in this attack is to replace blocks of an image with perceptually similar ones, obtained by geometrical and photometric transformations of a suitable block in the image. The attack can be applied to individual frames of a video sequence as well. The proposed countermeasure to BRA is to introduce spatial-coherency in the watermark such that similar areas in an image carry similar watermarks. Design of video watermarking schemes which embeds both spatial-and motion-coherent watermarks could be an interesting area of future research.

- **Perceptual quality metric:** Evaluating the perceptual distortion introduced by the watermarking process is a challenging problem. To date, video quality evaluation relies usually either on objective inaccurate metrics, e.g. the peak signal to noise ratio (PSNR), or on the detection of well-known visual artifacts, e.g. persisting pattern or temporal flicker [WGE03]. It could be possible to modify the proposed oracle to return a continuous value, indicating in some sense how much the embedded watermark flickers along the motion axis. This can be a useful tool to assess the perceptual impact of the watermark.

Bibliography

[Ada88] A. Adatia. Robust estimators of the 2-parameter Gamma distribution. *IEEE Transactions on Reliability*, 37(2):234–238, June 1988.

[ALC03] A. M. Alattar, E. T. Lin, and M. U. Celik. Digital watermarking of low bit-rate advanced simple profile MPEG-4 compressed video. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(8):787–800, August 2003.

[BBC05] M. Barni, F. Bartolini, and N. Checcacci. Watermarking of MPEG-4 video objects. *IEEE Transactions on Multimedia*, 7(1):23–32, February 2005.

[BBF02] M. Barni, F. Bartolini, and T. Furon. Security issues in digital watermarking. In *Proc. of 11th European Signal Processing Conference (EUSIPCO)*, volume 1, pages 282–352,441–461, September 2002.

[BBF03] M. Barni, F. Bartolini, and T. Furon. A general framework for robust watermarking security. *Signal Processing: Image Communication*, 83(10):2069–2084, October 2003.

[BCK+99] J. Bloom, I. Cox, T. Kalker, J.-P. Linnartz, M. Miller, and C. Traw. Copy protection for DVD video. *Proceedings of the IEEE*, 87(7):1267–1276, July 1999.

[BDP05] S. Biswas, S. R. Das, and E. M. Petriu. An adaptive compressed MPEG-2 video watermarking scheme. *IEEE Transactions on Instrumentation and Measurements*, 54(5):1853–1861, October 2005.

[BK04] U. Budhia and D. Kundur. Digital video steganalysis exploiting collusion sensitivity. In *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense III*, volume 5403 of *Proceedings of SPIE*, pages 210–221, April 2004.

[BKZ06] U. Budhia, D. Kundur, and T. Zourntos. Digital video steganalysis exploiting statistical visibility in the temporal domain. *IEEE Transactions on Information Forensics and Security*, 1(4):502–516, December 2006.

[BKZV05] N. Božinović, J. Konrad, W. Zhao, and C. Vázquez. On the importance of motion invertibility in MCTF/DWT video coding. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume II, pages 49–52, March 2005.

[BLD03] Y. Bodo, N. Laurent, and J.-L. Dugelay. Watermarking video, hierarchical embedding in motion vectors. In *Proceedings of the IEEE International Conference on Image Processing*, volume II, pages 739–742, September 2003.

[BPG05a] M. Barni and F. Pérez-Gonzàlez. Special session: watermarking security. In *Security, Steganography and Watermarking of Multimedia Contents VII*, volume 5681 of *Proceedings of SPIE*, pages 685–768, January 2005.

[BPG05b] M. Barni and F. Pérez-Gonzàlez. Tutorial: Security issues in digital watermarking. In *IEEE International Conference on Image Processing*, September 2005.

[BR02] R. Venkatesh Babu and K. R. Ramakrishnan. Background sprite generation using MPEG motion vectors. In *Proceedings of the Third Indian Conference on Computer Vision, Graphics and Image Processing*, pages 154–159, December 2002.

[BS87] K.O. Bowman and L. R. Shenton. *Properties of Estimators for the Gamma Distribution*. Marcel Dekker Inc., New York, 1987.

[BS98] D. Boneh and J. Shaw. Collusion secure fingerprinting for digital data. *IEEE Transaction on Information Theory*, 44(5):1897–1905, September 1998.

[BS04] A. Briassouli and M. G. Strintzis. Optimal watermark detection under quantization in the transform domain. *IEEE Journal on Circuits and Systems for Video Technology*, 14(12):1308–1319, December 2004.

[CFF05] F. Cayre, C. Fontaine, and T. Furon. Watermarking security: Theory and practice. *IEEE Transactions on Signal Processing, Supplement on Secure Media*, 53(10):3976–3987, October 2005.

[CM02] I. Cox and M. L. Miller. The first 50 years of electronic watermarking. *EURASIP Journal on Applied Signal Processing*, 2002(2):126–132, 2002.

[CMB01] I. Cox, M. Miller, and J. Bloom. *Digital Watermarking*. Morgan Kaufmann Publishers, 2001.

[CN05] P. CAMPISI and A. NERI. Perceptual video watermarking in the 3D-DWT domain using a multiplicative approach. In *Proceedings of the Fourth International Workshop on Digital Watermarking*, volume 3710 of *LNCS*, pages 432–443, September 2005.

[CPFPG05] P. Comesaña, L. Pérez-Freire, and F. Pérez-Gonzàlez. Fundamentals of data hiding security and their applications to spread spectrum analysis. In *Proceedings of the Seventh International Workshop on Information Hiding*, volume 3727 of *LNCS*, pages 146–160, June 2005.

[CST04] M. U. Celik, G. Sharma, and A. M. Tekalp. Collusion-resilient fingerprinting using random pre-warping. *IEEE Signal Processing Letters*, 11(10):831–835, October 2004.

[CW69] S. C. Choi and R. Wette. Maximum likelihood estimation of the parameters of the Gamma distribution and their bias. *Technometrics*, 11(4):683–689, November 1969.

[CW99] S.-J. Choi and J. Woods. Motion-compensated 3-D subband coding of video. *IEEE Transactions on Image Processing*, 8(2):155–167, February 1999.

[DCP00] F. Deguillaume, G. Csurka, and T. Pun. Countermeasures for unintentional and intentional video watermarking attacks. In *Security and Watermarking of Multimedia Contents II*, volume 3971 of *Proceedings of SPIE*, pages 346–357, January 2000.

[DCRP99] F. Deguillaume, G. Csurka, J. Ó Ruanaidh, and T. Pun. Robust 3D DFT video watermarking. In *Security and Watermarking of Multimedia Contents*, volume 3657 of *Proceedings of SPIE*, pages 113–124, January 1999.

[DD03a] G. Doërr and J.-L. Dugelay. A guide tour of video watermarking. *Signal Processing: Image Communication, Special Issue on Technologies for Image Security*, 18(4):263–282, April 2003.

[DD03b] G. Doërr and J.-L. Dugelay. New intra-video collusion attack using mosaicing. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, volume II, pages 505–508, July 2003.

[DD04a] G. Doërr and J.-L. Dugelay. Secure background watermarking based on video mosaicing. In *Security, Steganography and Watermarking of Multimedia Contents VI*, volume 5306 of *Proceedings of SPIE*, pages 304–314, January 2004.

[DD04b] G. Doërr and J.-L. Dugelay. Security pitfalls of frame-by-frame approaches to video watermarking. *IEEE Transactions on Signal Processing, Supplement on Secure Media*, 52(10):2955–2964, October 2004.

[DD05] G. Doërr and J.-L. Dugelay. Collusion issue in video watermarking. In *Security, Steganography and Watermarking of Multimedia Contents VII*, volume 5681 of *Proceedings of SPIE*, pages 685–696, January 2005.

[DDK06] G. Doërr, J.-L. Dugelay, and D. Kirovski. On the need for signal-coherent watermarks. *IEEE Transactions on Multimedia*, 8(5):896–904, October 2006.

[DHS01] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification – Second Edition*. Wiley-Interscience, 2001.

[Doë05] G. Doërr. *Security Issue and Collusion Attacks in Video Watermarking*. PhD thesis, Université de Nice Sophia-Antipolis, France, June 2005.

[Dop94] J. Dopke. Estimation of parameters in Gamma distribution. *International Journal of Quality and Reliability Management*, 11(8):27–43, October 1994.

[DS98] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.*, 4(3):245–267, 1998.

[DSR98] J. Dittman, M. Stabenau, and R.Steinmetz. Robust MPEG watermarking technologies. In *Proceedings of the Sixth ACM International Conference on Multimedia*, pages 71–80, 1998.

[EG01] J. J. Eggers and B. Girod. Quantization effects on digital watermarks. *Signal Processing: Image Communication*, 81(2):239–263, February 2001.

[FG99] J. Fridrich and M. Goljan. Comparing robustness of watermarking techniques. In *Security, Steganography and Watermarking of Multimedia Contents*, volume 3657 of *Proceedings of SPIE*, pages 214–225, January 1999.

[Fur05] T. Furon. A survey of watermarking security. In *Proceedings of the Fourth International Workshop on Digital Watermarking*, volume 3710 of *LNCS*, pages 201–215, September 2005.

[GF07] R. Givner-Forbes. Steganography: Information technology in the service of jihad. Technical report, The International Center for Political Violence and Terrorism Research, NTU Singapore, April 2007.

[GKF02] A. Gyaourova, C. Kamath, and I. K. Fodor. Undecimated wavelet transforms for image de-noising. Technical Report UCRL-ID-150931, LLNL Technical Report, 2002.

[HEG98] F. Hartung, P. Eisert, and B. Girod. Digital watermarking of MPEG-4 facial animation parameters. *Computers & Graphics*, 22(4):425–435, July 1998.

[HG98] F. Hartung and B. Girod. Watermarking of uncompressed and compressed video. *Signal Processing: Image Communication*, 66(3):283–301, May 1998.

[HK99] F. Hartung and M. Kutter. Multimedia watermarking techniques. *Proceedings of the IEEE*, 87(7):1079–1107, July 1999.

[HM05] Ö. Harmancı and K. Mıhçak. Motion picture watermarking via quantization of pseudo-random linear statistics. In *Visual Communications and Image Processing Conference*, volume 5960 of *Proceedings of SPIE*, pages 1142–1150, July 2005.

[HMT07] Ö. Harmancı, K. Mıhçak, and A. M. Tekalp. Watermarking and streaming compressed video. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume I, pages 833–836, April 2007.

[HMY00] M. Holliman, W. Macy, and M. Yeung. Robust frame-dependent video watermarking. In *Security and Watermarking of Multimedia Contents II*, volume 3971 of *Proceedings of SPIE*, pages 186–197, January 2000.

[HS05] E. Hauer and M. Steinebach. Robust digital watermark solution for intercoded frames of MPEG video data. In *Security and Watermarking of Multimedia Contents VII*, volume 5681 of *Proceedings of SPIE*, pages 381–390, January 2005.

[HW06] S. He and M. Wu. Collusion-resistant video fingerprinting for large user group. In *Proceedings of the IEEE International Conference on Image Processing*, pages 2297–2300, October 2006.

[IAH95] M. Irani, P. Anandan, and S. Hsu. Mosaic based representations of video sequences and their applications. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 605–611, June 1995.

[JDD99] R. C. Jones, D. DeMenthon, and D. S. Doermann. Building mosaics from video using mpeg motion vectors. In *Proceedings of the seventh ACM international conference on Multimedia (Part 2)*, pages 29 – 32, October 1999.

[JKE97] F. Jordan, M. Kutter, and T. Ebrahimi. Proposal of a watermarking technique for hiding/retrieving data in compressed and decompressed video. In *JTC1/SC29/WG11 MPEG97/M2281*. ISO/IEC, July 1997.

[Kal01] T. Kalker. Considerations on watermarking security. In *Proceedings of the IEEE Fourth Workshop on Multimedia Signal Processing*, pages 201–206, October 2001.

[KDHM99] T. Kalker, G. Depovere, J. Haitsma, and M. J. Maes. Video watermarking system for broadcast monitoring. In *Security and Watermarking of Multimedia Contents*, volume 3657 of *Proceedings of SPIE*, pages 103–112, January 1999.

[Kon04] J. Konrad. Transversal versus lifting approach to motion-compensated temporal discrete wavelet transform of image sequences: Equivalence and tradeoffs. In *Visual Communications and Image Processing Conference*, volume 5308 of *Proceedings of SPIE*, pages 452–463, January 2004.

[Kon05] J. Konrad. Importance of motion in motion-compensated temporal discrete wavelet transforms. In *Visual Communications and Image Processing Conference*, volume 5685 of *Proceedings of SPIE*, pages 354–365, January 2005.

[KP99] M. Kutter and F. A. P. Petitcolas. A fair benchmark for image watermarking systems. In *Security, Steganography and Watermarking of Multimedia Contents*, volume 3657 of *Proceedings of SPIE*, pages 226–239, January 1999.

[L. 00] L. Chiariglione . Short MPEG-2 description. In *JTC1/SC29/WG11 MPEG 00*. ISO/IEC, October 2000.

[Lan00] G. C. Langelaar. *Real-time Watermarking Techniques for Compressed Video Data*. PhD thesis, Delft University of Technology, Netherlands, February 2000.

[LD04] E. Lin and E. Delp. Temporal synchronization in video watermarking. *IEEE Transactions on Signal Processing*, 52(10):3007–3022, October 2004.

[LL01] G. C. Langelaar and R. L. Ladgendijk. Optimal differential energy watermarking of DCT encoded images and video. *IEEE Transactions on Image Processing*, 10(1):148–158, January 2001.

[LOL00] C.-H. Lee, H.-S. Oh, and H.-K. Lee. Adaptive video watermarking using motion information. In *Security and Watermarking of Multimedia Contents II*, volume 3971 of *Proceedings of SPIE*, pages 209–216, January 2000.

[LPKD01] E. Lin, C. Podilchuk, T. Kalker, and E. Delp. Streaming video and rate scalable video: What are the challenges for watermarking? In *Security and Watermarking of Multimedia Contents III*, volume 4314 of *Proceedings of SPIE*, pages 116–127, January 2001.

[Mat82] A. Mathai. Storage capacity of a dam with Gamma type inputs. *Annals of the Institute of Statistical Mathematics*, 34(1):591–597, December 1982.

[MH00] W. Macy and M. Holliman. Quality evaluation of watermarked video. In *Security and Watermarking of Multimedia Contents II*, volume 3971 of *Proceedings of SPIE*, pages 486–500, January 2000.

[MKR99] K. Mıhçak, I. Kozintsev, and K. Ramchandran. Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 3253–3256, March 1999.

[MKRM99] K. Mıhçak, I. Kozintsev, K. Ramchandran, and P. Moulin. Low-complexity image denoising based on statistical modeling of wavelet coefficients. *IEEE Signal Processing Letters*, 6(12):300–303, December 1999.

[MM05] Y. Mao and K. Mıhçak. Collusion-resistant intentional de-synchronization for digital video fingerprinting. In *Proceedings of the IEEE International Conference on Image Processing*, volume 1, pages 237–240, September 2005.

[MT03] N. Mehrseresht and D. Taubman. Adaptively weighted update steps in motion compensated lifting based scalable video compression. In *Proceedings of the IEEE International Conference on Image Processing*, volume III, pages 771–774, September 2003.

[NM07] M. Noorkami and R. M. Mersereau. A framework for robust watermarking of H.264-encoded video with controllable detection performance. *IEEE Transactions on Information Forensics and Security*, 2(1):14–23, March 2007.

[OdSW04] J. R. Ohm, M. Van der Schaar, and J. W. Woods. Inter-frame wavelet coding: Motion picture representation for universal scalability. *Signal Processing: Image Communication*, 19(9):877–908, October 2004.

[PAK98] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn. Attacks on copyright marking systems. In *Proceedings of the Second International Workshop on Information Hiding*, volume 1525 of *LNCS*, pages 218–238, April 1998.

[PF05] K. M. Parker and J. E. Fowler. Redundant-wavelet watermarking with pixel-wise masking. In *Proceedings of the IEEE International Conference on Image Processing*, volume I, pages 685–688, September 2005.

[PFCPG05] L. Pérez-Freire, P. Comesaña, and F. Pérez-Gonzàlez. Information-theoretic analysis of security in side-informed data hiding. In *Proceedings of the Seventh International Workshop on Information Hiding*, volume 3727 of *LNCS*, pages 131–145, June 2005.

[PGF05] F. Pérez-Gonzàlez and T. Furon. Watermarking security: where do we stand? In *Proceedings of the Fourth International Workshop on Digital Watermarking*, volume 3710 of *LNCS*, pages 201–274, September 2005.

[Pil97] M. Pilu. On using raw MPEG motion vectors to determine global camera motion. Technical Report HPL-97-102, Hewlett-Packard Research Laboratories, 1997.

[PTW06] R. Petrovic, B. Tehranchi, and J. M. Winograd. Digital watermarking security considerations. In *Proceedings of the ACM Multimedia and Security Workshop*, pages 152–157, September 2006.

[R. 06] R. Leonardi and A. Signoroni and S. Brangoulo. Status report-version 1 on wavelet video coding exploration. In *JTC1/SC29/WG11 MPEG2006/N7822*. ISO/IEC, January 2006.

[RS00] V. Rohatgi and E. Saleh. *An Introduction to Probability and Statistics*. Wiley-Interscience, 2000.

[Rup96] S. Rupley. What's holding up DVD? *PC Magazine*, 15(20):34–34, November 1996.

[Sch01] S. Schimmel. Motion sensitive video watermarking. Technical Report 2001/825, Philips Research, Eindhoven, The Netherlands, 2001.

[SH97] R. N. Strickland and H. I. Hahn. Wavelet transform methods for object detection and recovery. *IEEE Transactions on Image Processing*, 6(5):724–735, May 1997.

[She92] M. J. Shensa. The discrete wavelet transform: Wedding the À trous and mallat algorithms. *IEEE Transactions on Signal Processing*, 40(10):2464–2482, October 1992.

[SKH02] K. Su, D. Kundur, and D. Hatzinakos. A novel approach to collusion resistant video watermarking. In *Security and Watermarking of Multimedia Contents IV*, volume 4675 of *Proceedings of SPIE*, pages 491–502, January 2002.

[SKH05a] K. Su, D. Kundur, and D. Hatzinakos. Spatially localized image-dependent watermarking for statistical invisibility and collusion resistance. *IEEE Transactions on Multimedia*, 7(1):52–66, February 2005.

[SKH05b] K. Su, D. Kundur, and D. Hatzinakos. Statistical invisibility for collusion-resistant digital video watermarking. *IEEE Transactions on Multimedia*, 7(1):43–51, February 2005.

[SL01] I. Setyawan and R. Lagendijk. Low bit-rate video watermarking using temporally extended differential energy watermarking (DEW) algorithm. In *Security and Watermarking*

of Multimedia Contents III, volume 4314 of *Proceedings of SPIE*, pages 73–84, January 2001.

[ST03] A. Secker and D. Taubman. Lifting-based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression. *IEEE Transactions on Image Processing*, 12(12):1530–1542, December 2003.

[Sta06] W. Stallings. *Cryptography and Network Security(4/e)*. Prentice Hall, Inc., 2006.

[STB+04] D. Simitopoulos, S. A. Tsaftaris, N. V. Boulgouris, A. Briassouli, and M. G. Strintzis. Fast watermarking of MPEG-1/2 streams using compressed-domain perceptual embedding and a generalized correlator detector. *EURASIP Journal on Applied Signal Processing*, 2004(8):1088–1106, July 2004.

[SZT98] M. Swanson, B. Zhu, and A. Tewfik. Multiresolution scene-based video watermarking using perceptual models. *IEEE Journal on Selected Areas in Communications*, 16(4):540–550, May 1998.

[Tar03] G. Tardos. Optimal probabilistic fingerprint codes. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, volume II, pages 116–125, June 2003.

[TRS96] B. Tannenbaum, R.Suryadevara, and S.Hsu. Evaluation of a mosaic based approach to video compression. In *Proceedings of the Fifth International Conference on Computer Vision*, pages 1213–1215, May 1996.

[TSV+00] P. Termont, L. De Stycker, J. Vandewege, M. Op de Beeck, J. Haitsma, T. Kalker, M. Maes, and G. Depovere. How to achieve robustness against scaling in a real-time digital watermarking system for broadcast monitoring. In *Proceedings of the IEEE International Conference on Image Processing*, volume 1, pages 407–410, September 2000.

[Tud95] P. N. Tudor. MPEG-2 video compression. *IEE Electronics and Communication Engineering Journal*, 7(6):257–264, December 1995.

[vcd] VCDemo: Image and Video Compression Learning Tool. [online]. Available: http://ict.ewi.tudelft.nl/∼inald/vcdemo .

[VDPP01] S. Voloshynovskiy, F. Deguillaume, S. Pereira, and T. Pun. Optimal adaptive diversity watermarking with channel state estimation. In *Security and Watermarking of Multimedia Contents III*, volume 4314 of *Proceedings of SPIE*, pages 673–685, January 2001.

[VHS+07] A. L. Varna, S. He, A. Swaminathan, M. Wu, M. Lu, L. Haiming, and Zengxiang. Collusion-resistant fingerprinting for compressed multimedia signals. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume II, pages 165–168, April 2007.

[VPH+00] S. Voloshynovskiy, S. Pereira, A. Herrigel, N. Baumgärtner, and T. Pun. Generalized watermarking attack based on watermark estimation and perceptual remodulation. In *Security and Watermarking of Multimedia Contents II*, volume 3971 of *Proceedings of SPIE*, pages 358–370, January 2000.

[VPP+01] S. Voloshynovskiy, S. Pereira, T. Pun, J. J. Eggers, and J. K. Su. Attacks on digital watermarks: classification, estimation based attacks, and benchmarks. *IEEE Communications Magazine*, 39(8):118–126, August 2001.

[WCF03] Y. Wang, S. Cui, and J. E. Fowler. 3D video coding using redundant-wavelet multihypothesis and motion-compensated temporal filtering. In *Proceedings of the IEEE International Conference on Image Processing*, volume II, pages 755–758, September 2003.

[WGE03] S. Winkler, E. Gelasca, and T. Ebrahimi. Towards perceptual metrics for video watermark evaluation. In *Applications of Digital Image Processing*, volume 5203 of *Proceedings of SPIE*, pages 371–378, August 2003.

[WMS04] O. Werner, A. J Mason, and R. A. Salmon. Watermarking. Europian Patent (EP1465404), October 2004.

[WTWL04] M. Wu, W. Trappe, J. Wang, and R. Liu. Collusion-resistant fingerprinting for multimedia. *IEEE Signal Processing Magazine*, 21(2):15–27, March 2004.

[XL01] M. Xia and B. Liu. Effect of JPEG compression on image watermark detection. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume III, pages 1981–1984, May 2001.

[ZHQM07] J. Zhang, A.T.S. Ho, G. Qiu, and P. Marziliano. Robust video watermarking of H.264/AVC. *IEEE Transactions on Circuits and Systems (TCAS), Part II*, 54(2):205–209, February 2007.

[ZLZ01] J. Zhang, J. Li, and L. Zhang. Video watermark technique in motion vector. In *Proceedings of the 14th Brazilian Symposium on Computer Graphics and Image Processing*, pages 179–182, October 2001.

[ZWH04] L.-H. Zhang, H.-T. Wu, and C.-L. Hu. A video watermarking algorithm based on 3-D Gabor transform. *Journal of Software*, 15(8):1252–1258, August 2004.

[ZWWL05] M. V. Zhao, M. Wu, Z. J. Wang, and K.J.R Liu. Forensic analysis of nonlinear collusion attacks for multimedia fingerprinting. *IEEE Transactions on Image Processing*, 14(5):646–661, May 2005.