Robust Compressed Domain Video Watermarking for H.264/AVC

Tanima Dutta

Robust Compressed Domain Video Watermarking for H.264/AVC

Dissertation submitted in fulfillment of the requirements for the degree of

Doctor of Philosophy

by

Tanima Dutta

Under the guidance of

Dr. Arijit Sur

Department of Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
Guwahati 781039, India
2014

"Om Vang Me Manasi Pratisthita"

"May my mind be stable in my speech, May Atman manifest unto me and reveal unto me the Highest Knowledge"

-Aitareya Upanishad

In Memory of

Dadu, Thakurda and Thakuma

Dedicated to

Dida, Maa, Baba and Bhai

Whose blessings, constant inspiration and love made my path of success

Declaration

I certify that

1. The work contained in this dissertation is original and has been done by myself and the general supervision of my supervisor.

2. The work has not been submitted to any other Institute for any degree or diploma.

3. Whenever I have used materials (data, theoretical analysis, results) from other sources, I have given due credit to them by citing them in the text of the dissertation and giving their details in the references.

4. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Place: IIT Guwahati
Date:

Tanima Dutta
Research Scholar
Department of Computer Science and Engineering,
Indian Institute of Technology Guwahati,
Guwahati, Assam, INDIA 781039.

Certificate

*This is to certify that the work contained in this dissertation entitled "**Robust Compressed Domain Video Watermarking for H.264/AVC**" being submitted by **Tanima Dutta**, carried out in the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, is a bona fide research work of my supervision and is worthy of consideration for the award of the degree of Doctor of Philosophy of the Institute.*

Place: IIT Guwahati
Date:

Dr. Arijit Sur
Assistant Professor
Department of Computer Science and Engineering,
Indian Institute of Technology Guwahati,
Guwahati, Assam, INDIA 781039.

Acknowledgments

I would like to express my sincere gratitude to my supervisor Dr. Arijit Sur for his valuable guidance, inspiration and advice. His support and encouragement generously paved the way for my development as a research scientist. I benefited greatly from many fruitful discussions with him, which has changed my personality, ability and nature in many ways. I am highly grateful to my Doctoral Committee members, Dr. S.V. Rao, Prof. P. K. Bora and Dr. Pinaki Mitra. Their comments and suggestions have truly deepened and widened my understanding of the problems I have worked on. I am also deeply thankful Prof. G. Barua, Prof. S. Nandi and Dr. P.K. Das for their immense support, advice and encouragement. I express my sincere thanks to Dr. P. Bhaduri, the former Head, and Prof. S. B. Nair, the present Head of the Department of Computer Science and Engineering, for providing a nice research environment in the department, and support my research works in many ways.

I am thankful to TATA Consultancy Services, India for awarding me the research fellowship that gave me extremely good opportunities to broaden my research activities and interact with eminent researchers in the world, both from the industry as well as from the academia. I am grateful to Dr. P. Balamuralidharan, Principal Research Scientist, TCS Innovation Lab, Bangalore and Dr. A. Pal, Principal Research Scientist, TCS Innovation Lab, Kolkata for providing valuable comments and suggestions on my research outputs and giving a good exposure on the real world research challenges in the field of digital forensic, through interactions at TCS Innovation Labs. I thank Mr.

Rahul Pandey, former Program Manager and Mr. Sachin Parkhi, present Program Manager of TCS Research Scholar Program, for extending their helps and supports in technical and official activities.

I would also like to express my hearty gratitude to the Director, the Deans and other managements of IIT Guwahati whose collective efforts have made this institute a place for world-class studies and research. I am thankful to all the faculties and the staffs of the Department of Computer science and Engineering for extending their cooperation in terms of technical and official supports. I am obliged to the research scholars, M. Tech and B. Tech students of this institute with whom I have closely worked.

I am beholden to my friend Hari Prabhat, without whose help, motivation and support, the work might not be possible within this duration of time. Whenever I faced any difficulty, either in work or otherwise, he was beside me to come out of that. Our countless discussions and sharing of ideas and thoughts have helped us to debut most of the problems in research as well as in life, that surely accelerated both of our progresses. I also thank my childhood friend Payel, for constantly motivating and supporting me during my PhD days. I am indebted to all my friends, only to name a few, Moumita, Sumit, Shirsendu, Ruchira, Sathisha and Shaloni with whom I have enjoyed my student life. Last but not the least, I would like to express my gratitude to my parents and grandparents for their constant support and encouragement. Their motivation, assistance and guidance helped me to find my path for my future life. My brother deserves a special note of thanks for his enormous love and trust over me which helped me to overcome all the tough situations in life and will always inspire me to move forward towards my destination.

Place: IIT Guwahati

Date: **Tanima Dutta**

Abstract

In recent years, digital watermarking is being regarded as a promising solution in order to mitigate the increasing threats of video piracy and for ensuring video ownership and content authentication. Digital videos are generally stored and transmitted in a compressed format, which encourage research in designing video watermarking algorithm in compressed domain to avoid the complete decoding and re-encoding of a compressed video. H.264/AVC is a recent and an efficient video compression standard which provides higher compression than the MPEG-2 and MPEG-4, maintaining acceptable perceptual quality. The performance of video watermarking is often evaluated with parameters such as robustness, imperceptibility, security, transparency, payload, increase in video bit rate, blindness, etc. Most of these parameters are interdependent, often conflicting and are chosen based on the application. In this dissertation, different issues in compressed domain watermarking for H.264/AVC encoded video have been addressed. In this work, a few video frames (I-frame and P-frame) based watermarking algorithms have been designed and different challenges associated with them have been solved.

In the first part the dissertation, a P-frame based robust and blind watermarking method is proposed. The proposed method not only provides controlled increase in video bit rate and better perceptual quality of watermarked video, but can also withstand frame dropping, averaging and swapping (FDAS) attack through a robust watermarking method that uses repeat accumulate code and erasure channel. Syn-

chronization errors due to watermark embedding in compressed domain has also been minimized using an efficient block selection method. The proposed method uses a random key to select candidate blocks, which increases the security of the system. In the next part of the dissertation, a robust reversible watermarking method in P-frame with blind extraction process is proposed. The work ensures higher visual quality by selecting suitable blocks for embedding by analyzing spatial and temporal characteristics of the video and preventing distortion drift. Drift error propagation in intra and inter predicted macroblocks is avoided using the reversible watermarking technique. The proposed reversible watermarking algorithm is robust in nature. FDAS attacks have been also handled using reed-solomon codes.

In the third part of the dissertation, a robust watermarking algorithm for I-frame providing better visual quality and acceptable increase in the video bit rate is proposed. The drift error propagation in intra predicted blocks is handled. Keys are extracted to minimize the size of location map to prevent self collusion attack. An efficient block selection process decreases propagation of synchronization error. The extraction of watermark bits at the decoder using both location aware and unaware techniques is performed. A watermarking algorithm to resist collusion attack, which is based on motion compensated temporal frame averaging (MC-TFA), for compressed video is proposed in the last part of the dissertation. Embedding similar watermark in motion coherent homogeneous region can resist such collusion attacks. The motion coherent blocks are selected based on the pseudo motion vectors of I-frame to resist MC-TFA. Such blocks are grouped together based on the luminance prediction modes and the chrominance prediction modes. Motion coherent regions are merged into distinct clusters. The blocks in similar clusters are embedded with same watermark bits and blocks in different clusters are watermarked with different embedding bits. Finally, the dissertation concludes by briefly summarizing the work presented and discussing possible future research directions.

Contents

List of Figures

List of Tables

List of Symbols

$C(n)$	n^{th} coefficient in a 4×4 block in zigzag scan order
L	Length of the random watermark
l	Number of bits embedded per frame
F	total number of frames that are watermarked
q	RA code repetitions
H	Error correcting capability in any code
Z	Error correction capability using erasure channel
$\|z\|$	Absolute value of variable z
MVx	Motion vector in horizontal direction
MVy	Motion vector in vertical direction
MV_{th}	Motion threshold
$MASK_T$	Temporal Mask
$MASK_S$	Spatial Mask
NNZ_{th}	NNZ Threshold
R_T	Robustness Threshold
V_T	Visual Quality Threshold
S_{th}	Similarity Threshold
$Pr(X)$	Probability of the event X
W_i	i^{th} bit in original watermark sequence W
W_i^*	i^{th} bit in extracted watermark sequence W^*

K	Key
$P(z)$	Luminance intra prediction mode of block z
$M(z)$	Chrominance prediction mode of block z
f	Current frame number
k	GOP length

Abbreviations

FDAS	Frame dropping, averaging, and swapping
GOP	Group of pictures
MC-TFA	Motion compensated temporal filtering averaging
BIR	Bit Increase Rate
BER	Bit Error Rate
PSNR	Peak Signal-to-Noise Ratio
VQM	Visual Quality Metric
RA	Repeat-Accumulate
RS	Reed-Solomon
AC	AC coefficient
DC	DC coefficient
LSB	Least significant bit
NNZ	Number of Nonzero coefficients in a 4×4 block
chroma	Chrominance prediction mode
luma	Luminance prediction mode
block	4×4 block
macroblock	16×16 block

Chapter 1

Introduction

Digital media has been accepted as an ubiquitous means of storage and communication of electronic data since last two decades because of their notable benefits in efficient storage, ease of manipulation, and less transmission overhead. It is however unfortunate that the very nature of digital media itself has made the work of the intruders and hackers a bit easier to make a perfect copy with no loss of value. This has been a serious threat to the digital media producers and copyright owners to protect media from a potential intruder to avoid the loss in business.

Digital watermarking is a tool to embed a digital signature into digital media [GBW01, AS07, AYCK04]. Digital watermarking is suitable for different media forensics applications like ownership authentication against digital piracy, media content authentication against media forgery, traitor tracing, broadcast monitoring etc. For ownership authentication, a robust signal is embedded into the cover signal (image, video etc. to be authenticated) such that the embedded signal can be extracted at the receiver side to authenticate the ownership of the cover media. The watermark does not prevent a user from listening to, viewing, examining, or manipulating the content. The media could contain information in the form of images, video, audio, etc.

Digital video is widely used and distributed in electronic media. The importance

of digital video has come into limelight with the emergence of digital television, digital versatile disks (DVD), and video transmission over the Internet. Video watermarking is considered as a promising solution for content authentication and copy protection. A brief overview of video watermarking is described in the next section.

1.1 Video Watermarking

Video watermarking is a technique to embed a digital signature into the video stream for copyright protection [EK01]. The model of video watermarking algorithm is described in Figure 1.1. A video is watermarked using a watermarked signal and a secret key. The watermarked video is sent over communication channels to the client. Attackers may attack and channel noises may add during the transmission. An authorized client extracts the watermark from the watermarked video to authenticate the ownership using the secret key. Video watermarking is used in a wide range of applications, such as, content authentication, transactional watermarks (fingerprinting), broadcast monitoring, etc.

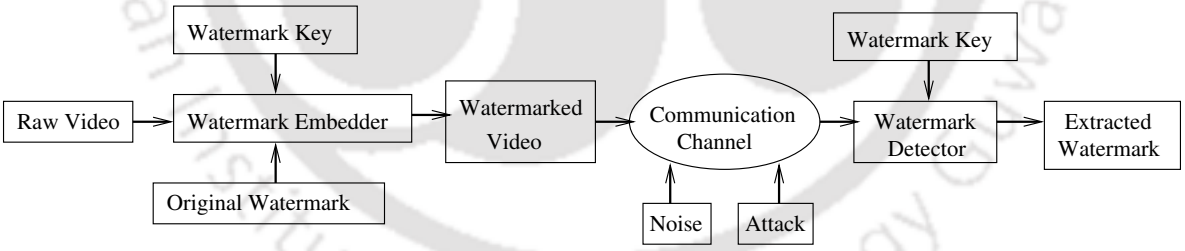


Figure 1.1 Block diagram of Video Watermarking [Ric10].

1.1.1 Video Watermarking Applications

With the wide applications of digital video, watermarking can add value to various video applications. Video watermarking applications are presented extensively in [DD03]. Some of these applications are summarized as follows.

Applications of video watermarking	
Application	Purpose of the embedded watermark
Copy control	Prevent unauthorized copying.
Source tracking	Different recipients get differently watermarked content.
Broadcast monitoring	Identify the video being broadcast and check usage.
Fingerprinting	Trace back a malicious user.
Video authentication	Insure that the original content has not been altered.
Covert communication	Hiding the fact that anything sensitive exists at all.
Copy protection	Prove ownership.
Advertisement	Verify the frequency of display of an advertisement.

1.1.2 Video Watermarking Parameters

A video watermarking algorithm can be evaluated using different performance measures, which are listed as follows:

- *Robustness* can be defined as how efficiently a watermarking system can resist different innocent or malicious attacks on the video. For example, the innocent content processing are compressions, filtering, noise addition, etc., while different malicious attacks are geometric attacks, frame dropping, averaging, and swapping attacks, watermark estimation attacks, etc.

- Watermarking *fidelity* measures the degradation of the perceptual quality of the watermarked video stream due to the watermarking process. Fidelity is the measure of the similarity between video stream before and after processing. Fidelity in the watermarking applications is the principal perceptual measure of concern.

- The *payload* is the amount of watermark signal (*e.g.* number of bits) that can be embedded in a video. Payloads of watermarks can vary from one bit of information (which typically indicates whether the image contains a specified watermark) to

several bytes of information.

- Watermarking *security* is defined as the resistance to the unauthorized users to access (i.e. remove, read, or write) the hidden message, in the communication channel established by a robust watermarking. Security has to do with the ability of the watermarking systems to properly conceal the information that must be kept secret, such as, the secret keys and the embedded messages.

While designing a watermarking algorithm, it is observed that the trade-offs exist among these parameters: payload, fidelity, security, and robustness. These parameters however are chosen based on the applications.

1.1.3 Classifications of Video Watermarking Techniques

Video watermarking techniques can be classified into three distinct categories as follows:

- **Uncompressed domain vs Compressed domain**: In *uncompressed domain* video watermarking, the watermark is embedded into the raw video. The uncompressed domain video watermarking methods with compressed video streams require full decoding and re-encoding which is computationally intense process and thus not very useful in the real time video processing.

  In *compressed domain* watermarking, compressed domain features (like motion vector, quantized DCT of the block residual coefficients, prediction mode, etc.) are modified to embed watermark. The complete decoding and re-encoding of the video sequence is not required to embed watermark bits in compressed domain. The error introduced by watermark embedding in compressed domain however propagates into subsequent frames and thus maintaining an acceptable visual quality of the watermarked video is one of the key challenges.

- **Blind vs Non-blind**: *Blind* video watermarking algorithms do not require the original video to detect the watermark at the decoder end. *Non-blind* watermark-

ing algorithms do require the original video to extract / detect the watermark. The designing of a video watermarking system as blind or non-blind depends on the application.

- **Fragile, Semi-fragile, and Robust**

  1. A digital watermark is called *fragile* if it fails to be detectable after the slightest modification of watermarked video. Fragile watermarks are commonly used for tamper detection (integrity proof) for legal, military, and medical imaging applications.
  2. A digital watermark is called *semi-fragile* if it resists some intentional transformations. Semi-fragile watermarks are mostly used as content dependent watermarks where content needs to be strictly protected, but the exact representation during communication and storage need not be guaranteed.
  3. A digital watermark is called *robust* if the embedded watermark may be detected reliably from the watermarked video, even if degraded by a designated class of transformations. Robust watermarks may be used in copy protection applications to carry ownership authentication and access control.

1.1.4 Video Watermarking Attacks

The robustness of a video watermarking algorithm are evaluated against different unintentional noise like channel noise, synchronization error, etc. and intentional attacks like image processing attacks, watermark estimation attack, etc. A brief description of few popular attacks are as follows:

- **Image Processing Attack**: Some image processing attacks that are commonly used are like noise addition, such as, salt and pepper noise and gaussian noise, filtering using gaussian filter and average circular filter, and intentional attacks include cropping attack and rotation-scaling-translation attack [NM07, MAAK10].

- **Watermark Estimation Attack**: A watermark estimation attack is usually accomplished by means of a denoising procedure. The common step used to realize a collusion or copy attack is the estimation of watermark bits [DD03, LHC06]. In collusion attack, watermarked video frames are analyzed or combined with the goal of removing the watermark. Two types of inter-frame collusion attacks are possible: different frames with same watermark, *i.e.*, Type I collusion attack and copies of the same frame with different watermarks, *i.e.*, Type II collusion attack. Inter-frame collusion attacks exploit the inherent redundancy within the video frames or in watermark to produce the unwatermarked copy of the video [PDB09].

- **Swap Attack**: A swap attack locates perceptually similar regions of a video and copies one such region to another [HM00, KP03]. The fact that the video has highly repetitive phenomena that spurred a successful generation of swap attacks, which replace relatively lengthy watermarked blocks of video with perceptually similar blocks found elsewhere and hence, the corresponding watermark is removed. Frame dropping, averaging, and swapping (FDAS) attack is a typical swap attack for videos.

- **Synchronization Error**: When the watermark signal is embedded in a compressed video, error occurs due to loss of synchronization which decreases the robustness of the video watermarking algorithm and degrades the visual quality [MAAK10]. The synchronization error is more in case of highly compressed video like H.264/AVC. Due to embedding, the prediction modes are changed after re-encoding, which affects the synchronization in the watermark extraction process even with same configuration parameters [MAAK10]. Changing configuration parameters (re-encoding attack [MAAK10]) or even recompression (recompression error) increases desynchronization in the watermark extraction process at the decoder. The prediction error residuals, motion vectors, and reference blocks may

be changed due to changes in prediction modes, which decreases the watermark detection rate.

H.264/Advanced Video Coding (AVC) is regarded as one of the efficient industry standards for video coding. It is built on the concepts of MPEG-2 and MPEG-4. In this dissertation, the H.264 compressed videos are used for experimentations. A brief overview of the encoder and decoder of H.264/AVC is presented in the next section.

1.2 H.264/AVC Basics

H.264/AVC is a popular format for coded video and a set of tools for video compression. It is co-published by two international standard bodies, *i.e.*, ITU-T (International Telecommunication Union) and ISO/IEC (International Organization for Standardization/International Electro-technical Commission). In Figure 1.2, the block diagram of H.264/AVC encoder and decoder is depicted. In this section, a brief description of encoding and decoding process for H.264/AVC standard is presented.

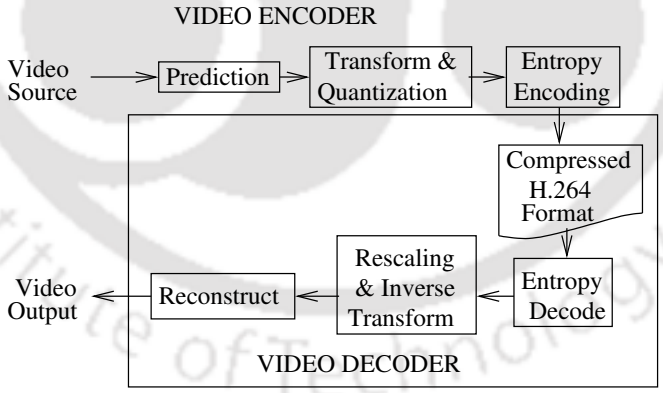


Figure 1.2 Block diagram of H.264/AVC Encoder/Decoder [Ric10].

1.2.1 GOP Structure

In Figure 1.3 the Group of Pictures (GOP) structure in H.264/AVC is depicted. In a GOP, three types of frames are possible in H.264/AVC, *i.e.*,

1. Intra frame, denoted by I-frame, are also known as key frame. I-frame is least compressed but do not require other video frames to decode and every GOP contains one I-frame.

2. Predicted frame, denoted by P-frame, is unidirectionally predicted from previous frame. Generally P fames are more compressed than the I frames.

3. B-frame, which are bidirectionally predicted from both previous and forward frames for data reference to get the highest amount of data compression.

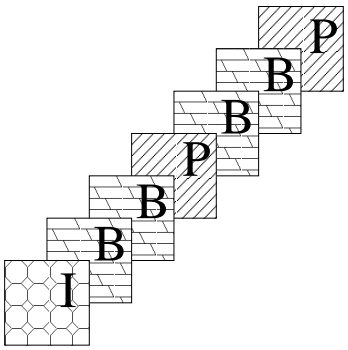


Figure 1.3 Example of a Group of Picture (GOP) structure, where length of the GOP is 7 [Ric10].

In H.264/AVC compression standard, generally one I-frame occurs in a GOP. Mostly in highly compressed video, the number of P-frames are more than I-frames and the number of B-frames is more than P-frames. However, the structure of the GOP can be varied.

1.2.2 Encoding Process

A brief architecture of H.264/AVC encoder is depicted in Figure 1.4. The operations that H.264/AVC encoder performs to compress the raw video, such as, prediction, transformation, quantization, and bitstream encoding, etc. are briefly described in this section.

Prediction

The encoder predicts the current macroblock based on previously coded macroblocks, either from the current frame using intra prediction or from previous or future frames that have already been coded and transmitted using inter prediction as illustrated in Figure 1.5. The encoder subtracts the predicted macroblock from the current mac-

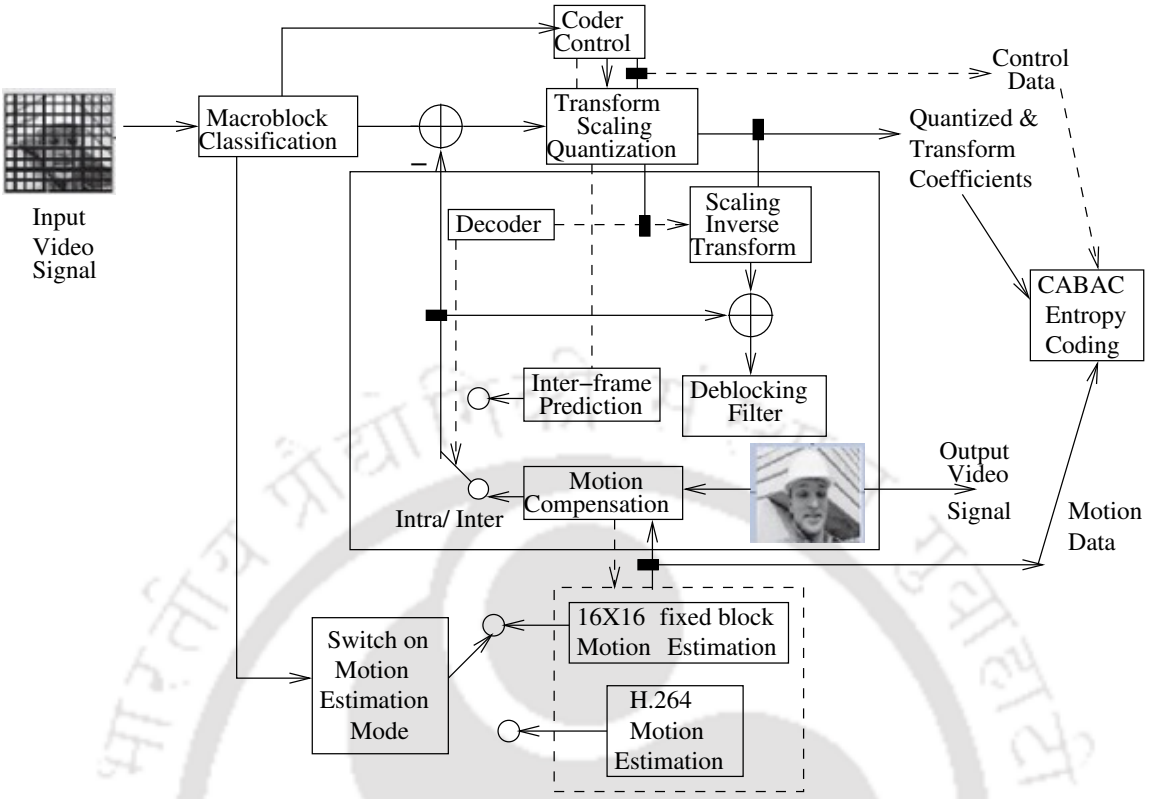


Figure 1.4 Architecture of H.264/AVC Encoder [MWS06].

roblock to form a residual macroblock. Intra prediction uses 16×16 and 4×4 block sizes to predict the macroblock from surrounding, previously coded pixels within the same frame as shown in Figure 1.6. Different luminance intra prediction modes for 4×4 block size are depicted in Figure 1.8. The luminance intra prediction modes for 16×16 block size and chrominance prediction modes for 8×8 block size are shown in Figure 1.7. Inter prediction modes as presented in Figure 1.9 uses a range of block sizes starting from 16×16 down to 4×4 to predict pixels in the current frame from previously coded frames as illustrated in Figure 1.10.

Transform and Quantization

A residual block is transformed using a 4×4 or 8×8 integer transform, which is an approximation of Discrete Cosine Transform (DCT). The transform generates a set of coefficients. These transform coefficients are quantized, *i.e.*, each coefficient is divided

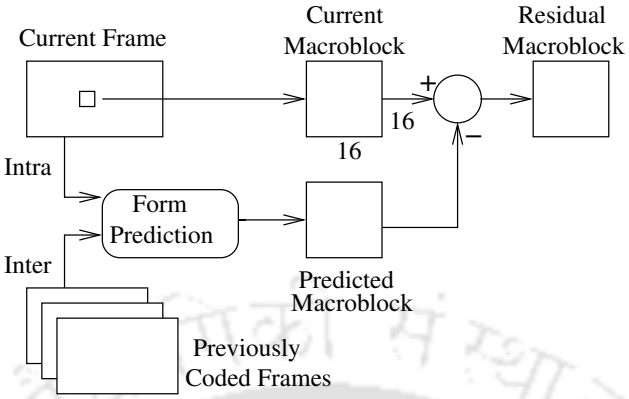


Figure 1.5 Prediction flow diagram [Ric10].

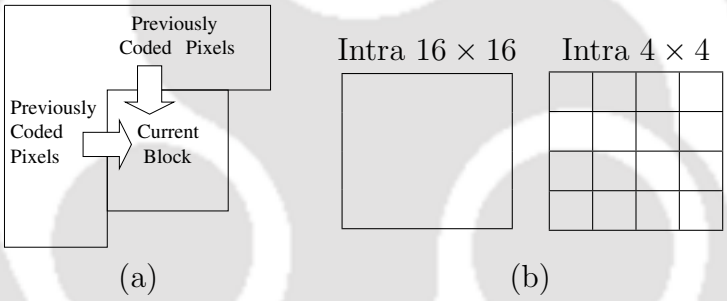


Figure 1.6 The intra prediction process and different luminance intra prediction block sizes are shown in part (a) and part (b) of the figure, respectively [Ric10].

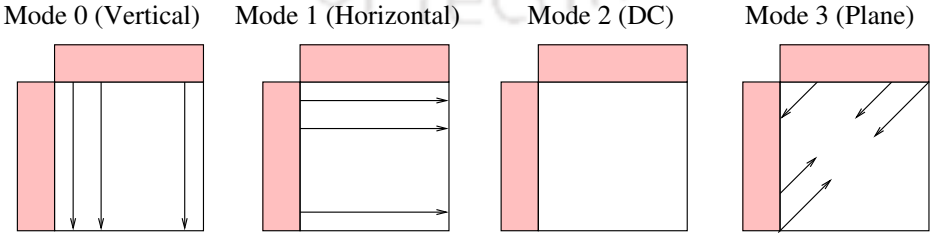
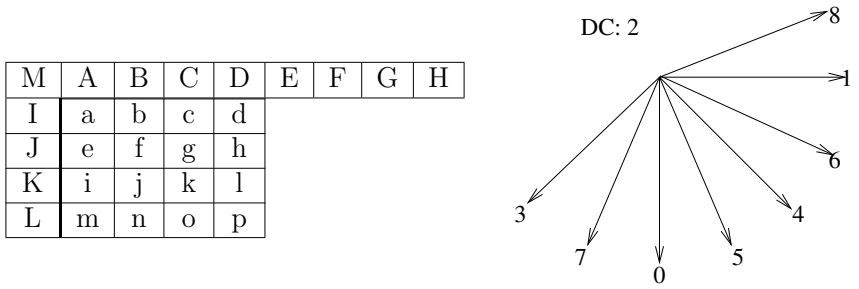


Figure 1.7 Luminance intra prediction mode for 16 × 16 blocks and chrominance prediction mode for 8 × 8 blocks [Ric10].



Mode	Name
0	Vertical
1	Horizontal
2	DC
3	Diagonal Down-Left
4	Diagonal Down-Right
5	Vertical-Left
6	Horizontal-Down
7	Vertical-Right
8	Horizontal-Up

Figure 1.8 Luminance intra prediction modes of 4×4 blocks [Ric10].

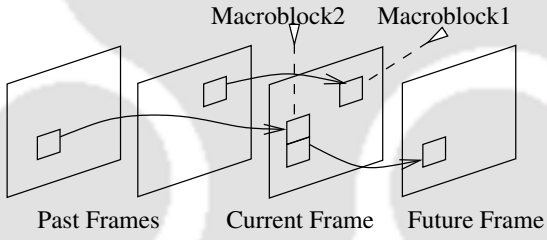


Figure 1.9 Inter Prediction Process [Ric10].

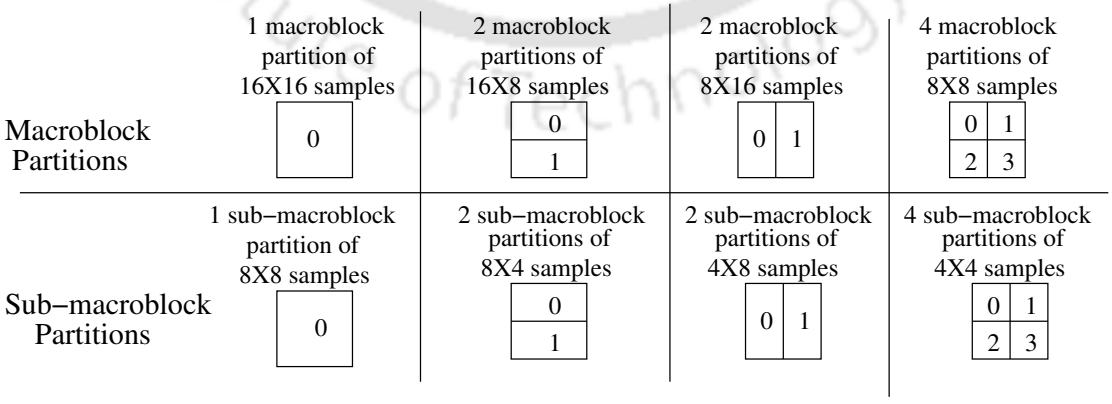


Figure 1.10 Different inter prediction block sizes [Ric10].

by an integer value and rounded to nearest integer. The purpose of quantization process is to reduce the precision of the transform coefficients according to a quantization parameter, denoted by QP. If QP is set to a higher value, then, most of the coefficients are set to zero resulting in high compression at the cost of poor decoded video quality [Ric10]. Similarly, setting QP to a low value keeps more nonzero coefficients after quantization, which results in better video quality at the decoder but lower compression.

Bitstream Encoding

The video coding process generates a number of values, such as, quantized transform coefficients, information that enable the decoder to perform the prediction process, information about the structure of the compressed data and the compression tools used for encoding, and information about the complete video sequence [Ric10]. These values, encoding parameters and syntax elements are converted into binary codes using variable length coding and/or arithmetic coding to form the compressed bitstream. All these encoding algorithms produce an efficient and compact binary representation of the information. The encoded bitstream is then stored and/or transmitted.

1.2.3 Decoding Process

H.264/AVC decoder converts a compressed video stream back into an uncompressed format. The decoding process performs bitstream decoding, rescaling and inverse transformation, reconstruction, etc.

Bitstream Decoding

A video decoder, after receiving the compressed H.264 bitstream, decodes the syntax elements and extracts the information, *i.e.*, quantized transform coefficients, prediction information, and compression details. The extracted information is then used to reverse the coding process to recreate the video sequence.

Rescaling and Inverse Transform

Each of the quantized coefficient is rescaled by multiplying with an integer value (quantization parameter) to restore its original scale. The reconstructed residual blocks, obtained after the inverse transform, however are similar but not identical to the original residual block as quantization is a lossy process. Figure 1.11 shows the reconstruction flow diagram.

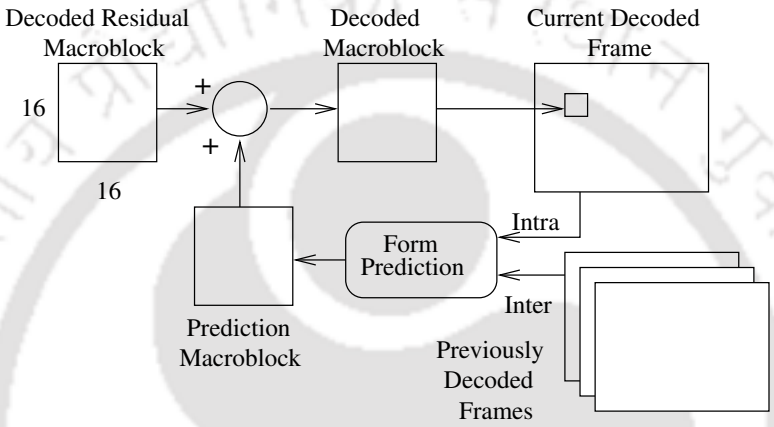


Figure 1.11 Reconstruction flow diagram [Ric10].

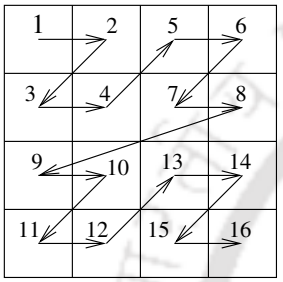
Reconstruction

The decoder performs prediction for each macroblock by adding the predicted macroblock to the decoded residual macroblock to reconstruct a decoded macroblock using inter prediction from previously decoded frames or intra prediction from previously decoded blocks in the current frame. The decoded macroblock is then displayed as part of a video frame as depicted in Figure 1.11.

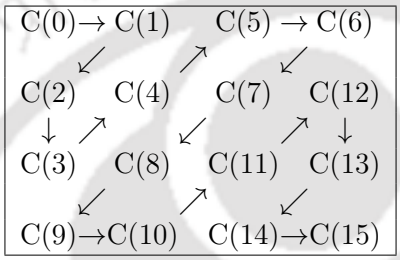
1.2.4 Zigzag Scan Order

Figure 1.12 depicts 4×4 blocks in a macroblock and coefficients in a 4×4 block, which are encoded in zigzag scan order. The n^{th} coefficient, DC coefficient and AC coefficients in a 4×4 block are denoted by $C(n)$, $C(n = 0)$, and $C(n \neq 0)$, respectively. AC coefficients

can be divided into two sub-sequences, namely, odd sequence and even sequence. Odd and even is identified using the index value n in Figure 1.12(b). Coefficients in the odd sequence in Figure 1.12(c) are AC(0,1), AC(2,0), AC(0,2), AC(1,2), AC(3,0), AC(2,2), AC(2,3), and AC(3,3). Similarly, coefficients in the even sequence in Figure 1.12(c) are AC(1,0), AC(1,1), AC(0,3), AC(2,1), AC(3,1), AC(1,3), and AC(3,2). Colored cells denote coefficients in the odd sequence while remaining cells are in the even sequence (except DC).



(a)



(b)

DC	AC(0,1)	AC(0,2)	AC(0,3)
AC(1,0)	AC(1,1)	AC(1,2)	AC(1,3)
AC(2,0)	AC(2,1)	AC(2,2)	AC(2,3)
AC(3,0)	AC(3,1)	AC(3,2)	AC(3,3)

(c)

Figure 1.12 Part (a) and (b) of the figure depict the zigzag scan order of 4×4 blocks in a macroblock and coefficients in a 4×4 block, respectively [Ric10]. Part (c) of the figure shows coefficients in odd and even sequences in a block.

1.3 Contribution of the Dissertation

In this section, the contribution of the dissertation is presented to resolve some aforesaid issues. The dissertation has four main parts. In this section, the contribution of each part is described.

1.3.1 Robust Watermarking in P-Frame

In this part, a novel P-frames based watermark embedding algorithm with blind extraction is proposed. The embedding algorithm is made robust to minimize synchronization error by appropriate selection of 4×4 blocks based on spatial and temporal characteristics of partially decoded video parameters. In P-frames, luminance intra and inter

prediction modes coexist. Based on luminance inter prediction modes, different block sizes are determined. There are 10 different prediction modes, *i.e.*, $\{4\times4$, 4×8, 8×4, 8×8, 8×16, 16×8, 16×16, SKIP$\}$, in P-frames. Selecting blocks based only on modes in P-frame may not be appropriate as prediction modes are changed frequently due to re-encoding. Watermark bits are embedded by modifying nonzero quantized coefficients to restrict the increase in video bit rate and provide better perceptual quality of the watermarked video. P-frame embedding method are rarely prone to FDAS attack. To prevent FDAS attack repeat-accumulate codes with an erasure channel is used to design a robust watermark. The proposed algorithm is robust to the image processing attacks.

1.3.2 Drift Compensated Watermarking in P-frame

A crucial problem of decoder based compressed domain watermarking algorithms is drift error propagation, which refers to the watermark error accumulations among different blocks during intra or inter predictions. In addition, there is a loss of synchronization in watermark extraction process due to embedding distortion. This desynchronization is enhanced in decoder based compressed domain watermarking methods. In this chapter, a robust and blind watermarking method is proposed which can handle the drift error propagation using the proposed robust reversible watermarking algorithm in P-frame. Different intra and inter prediction modes coexist and motion estimation is performed based on reference frames. A small modification in a 4×4 block may propagate drift distortion error to a number of blocks in that frame or frames predicted from the modified block degrading the visual quality of the watermarked video. Reversible watermarking allows to get back the original (unwatermarked) video at the decoder. The existing algorithms on reversible watermarking are fragile in nature. Therefore, a robust reversible watermarking algorithm is proposed. Realization of different thresholds is performed to maintain higher perceptual quality and robustness. The proposed algorithm is secured

using a random key. To prevent from FDAS attack, the error correcting capability of reed-solomon code is exploited.

1.3.3 Robust Watermarking in I-frame

In this work, a robust watermarking algorithm with blind extraction process is proposed. It embeds watermark invisibly in I-frame. The hybrid algorithm uses the advantages of the state of the art literature and removes the pitfalls, such as, perceptual distortion due to I-frame embedding, fragility, and distortion drift. Public keys are extracted using robust compressed domain features to minimize the location map. This decreases the overhead of secure transmission of large location maps. Moreover, the self collusion attack is also resisted using extracted keys. Appropriate selection of blocks decreases synchronization error in I-frame. Location aware and unaware extraction of watermark bits at the decoder are performed and compared with the existing literature

1.3.4 CRMC: Collusion Resistant Motion Coherent Watermarking

In the final part, a novel motion coherent watermarking algorithm is proposed that can resist collusion attack. To resist collusion attack, similar watermark must be embedded in similar region. First the motion coherent blocks are detected based on the pseudo motion vectors of I-frame. Such blocks are grouped together based on luminance prediction modes and chrominance modes. Motion coherent regions are merged into distinct clusters using the proposed clustering method. Then, blocks in similar clusters are embedded with same watermark bits and blocks in different clusters are watermarked with different embedding bits. This makes the watermarking method robust to inter-frame collusion attack and MC-TFA attacks. Blocks are detected based on motion coherency so frame averaging will not be able to remove the embedding bits. Therefore, a robust watermarking framework is designed such attacks.

1.4 Dissertation Organization

This dissertation is organized as follows.

- In Chapter 1, video watermarking concepts and basics of H.264 compression standard are introduced. The objective and the main contributions of the dissertation are also narrated.

- Chapter 2 briefly narrates the state of the art literature of video watermarking and summarizes the existing literature for watermarking in H.264/AVC encoded videos.

- A robust watermarking algorithm in P-frame for better visual quality with a controlled increase in the video bit rate is presented in Chapter 3. The analysis of the different frames of compressed video is performed to understand the change in perceptual quality of the watermarked videos based on different watermarking algorithms.

- A robust reversible watermarking method to compensate drift error propagation in intra and inter predicted blocks for embedding in P-frame of compressed video is introduced in Chapter 4. The synchronization error is also minimized.

- A robust watermark embedding algorithm in I-frame that minimized the size of location map and compensate drift error propagation for intra predicted blocks is presented in Chapter 5. The compressed domain watermarking algorithm enhances the embedding capacity with minimum perceptual distortion and prevents self collusion attack.

- A motion coherent compressed domain watermarking algorithm to resist collusion attacks (motion compensated temporal frame averaging) is introduced in Chapter 6. The method embeds the same watermark in similar motion coherent blocks to resist the collusion (MC-TFA) attack.

- The dissertation is finally concluded, future research scopes are suggested in Chapter 7. An application scenario of the work in the thesis is also illustrated.

Chapter 2

Literature Survey

In this chapter, a brief description of existing literature related to digital video watermarking is presented. The digital watermarking methods in video can be divided into three main groups:

- methods based on watermarking in still images,
- methods based on temporal dimension of video,
- methods based on video compression standards.

Video content is considered as a sequence of still images (frames). Therefore, a simple and straightforward approach is to reuse existing watermarking schemes for still images in a frame-by-frame fashion as described in Section 2.1. Alternatively, as presented in Section 2.2, the temporal dimension can be integrated in the watermarking procedure. In practice, this can be implemented very simply by considering video content as a collection of signal samples by exploiting for instance 3-D signal transforms. The last approach relies on the observation that video content is usually compressed with a specific video compression standard for storage/transmission convenience. As a result, Section 2.3 briefly describes different ways of exploiting such standards to obtain very efficient video watermarking schemes.

2.1 Methods based on Watermarking in Still Images

Video watermarking has been extensively investigated exclusively for still images in the past and many interesting results and algorithms were found. With the advance in the technology, when new areas, such as video, were researched upon, the basic concern was to try to reuse the previously found results. As a result the watermarking community first considered digital video content as a succession of still images (frames) and adapted existing image watermarking schemes to the video in a frame-by-frame fashion. Similarly, the same phenomenon occurred when the coding community switched from image coding to video coding. Indeed, the first proposed algorithm for video coding was Moving Joint Photographic Experts Group (M-JPEG), which simply compresses each frame of the video with the image compression standard JPEG. The simplest way of extending a watermarking scheme for still images is to embed the same watermark in the frames of the video. At the decoder, the presence of the watermark is checked in every frame in the same way as detected in still images. However, the main drawback of such a scheme is that it does not give any information about the payload *i.e.* the detector only tells if a given watermark is present or not but it does not extract any hidden message.

Since, video content is much larger in size than a single still image, one should be able to hide more watermark bits and high payload watermarks for video could be expected. To achieve this goal, the algorithm for Just Another Watermarking System (JAWS) has been designed, where the same message is embedded in all the frames of the video [KDHM99]. An $M \times M$ normally distributed reference pattern p_r is generated with a secret key. Then, a reference watermark w_r is created, such that $w_r = p_r - \text{shift}(p_r, m)$, where the shift(.) function returns a cyclically shifted version of the reference pattern p_r and m is the binary watermark to be hidden. This reference watermark w_r is then tiled, possibly with truncation, to obtain the full-size watermark w. For each frame, this watermark is then perceptually shaped so that the watermark insertion remains

imperceptible. Each element i of the watermark is scaled by the local activity (i) of the frame, such as Laplacian filtering and added to the frame to obtain the watermarked frame. At the decoder, the frames are folded, summed, and stored in an $M \times M$ buffer b. The decoder looks for all the occurrences of the reference pattern p_r in the buffer with a two dimensional cyclic convolution. Since such an operation is most efficiently computed in the frequency domain, this leads to Symmetrical Phase Only Matched Filtering (SPOMF) detection which is given by the following equation:

$$\text{SPOMF}(b, p_r) = \text{IFFT}\left[\emptyset\left(\text{FFT}(b)\right) \times \emptyset\left(\text{FFT}(p_r^*)\right)\right] \text{ with } \emptyset = \begin{cases} \frac{x}{|x|} & \textbf{if } x \neq 0 \\ 1 & \textbf{if } x = 0 \end{cases}, \quad (2.1)$$

where FFT(.) and IFFT(.) denotes the forward and inverse Fast Fourier Transform, respectively and x the complex conjugation of x. Once the decoder has extracted the peaks, the hidden payload can be easily retrieved according to the estimated shift. This scheme is inherently shift invariant since a shifting operation does not modify the relative position of the peaks. The shift invariance has been further exploited significantly to increase the payload [MKHD99] and simple modifications enabled to obtain scale invariance [TDSV$^+$00]. In [DSS98], an independent multi-bits watermark is embedded in each frame of the video to exploit the whole available bandwidth. In this case, the gain in embedding capacity is counterbalanced by a loss of robustness since each watermark bit is spread on fewer samples, which increased sensibility against desynchronization. In addition, frame by frame video watermarking may be prone to collusion attack.

2.2 Methods based on Temporal Dimension of Video

The major shortcoming of considering video content as a succession of independent still images is that the temporal dimension of the video is not satisfactorily taken

into account. On the other side, the coding community made a big step forward by incorporating motion prediction. Perceptual shaping is another issue which highlights the fact that the temporal dimension is a crucial point in video and that it is not taken into account to design efficient algorithms. Many researchers have investigated that the obtained watermark using still image watermarking in videos is not optimal in terms of visibility since it does not consider the temporal sensitivity of the human eye. Some of the pioneer works in video watermarking considers video content as a one dimensional signal, such as Spread Spectrum watermarking [HG98]. In other words, such algorithms discard any notion of dimensionality, whether it be spatial or temporal, and looks at the video signal as a collection of signal samples.

In [HG98], the bipolar watermark sequence (-1,1) is spread to add redundancy by embedding one bit of information into samples of the video signal, scaled locally to adjust the spatial and temporal masking of the Human Visual System (HVS), and modulated by a pseudo-random binary sequence ($\in \{1, 1\}$). Finally, the obtained spread spectrum watermark is added to obtain the watermarked video signal. At the decoder side, recovery is accomplished with a simple correlation. However, to reduce cross-talk between watermark and video signals, the watermarked video signal is high-pass filtered, yielding a filtered watermarked, so that major components of the video signal itself are isolated video signal and removed. Next, the filtered watermarked video signal is multiplied by the pseudo-random noise pattern, which is used for embedding and summed over the window for each watermark bit. The hidden bit is finally extracted from the sign of correlation sum. This method however completely ignores spatio-temporal dimensions. Therefore, the resulting embedded watermark is likely not to be optimal in terms of invisibility and robustness. Different approaches have consequently been investigated how to insert a temporal watermark sequence at some key-dependent specific pixel locations [MSB02]. To ensure watermark invisibility, the embedding locations have to be carefully chosen, for example if modifying a single pixel in a textured

area is imperceptible in each individual video frame, it might become visible when the video is rendered.

Many researchers have investigated that pixels which change fast along the time axis or pixels in border areas of motionless regions have been shown to be good candidates for embedding. Nevertheless, using only a few pixels for watermark embedding drastically reduces the embedding capacity. In fact, one may prefer to compute some temporal transform on the whole video to have a larger embedding space. In particular, temporal wavelet decomposition can be useful to obtain a compact multi-resolution temporal representation of the video [SZT98]. With such a decomposition, one can isolate a static (no motion) component and several dynamic (motion) ones. The multi-resolution nature of the wavelet transform allows the watermark to exist across multiple temporal scales. For instance, if a watermark is embedded in the lowest temporal frequency (DC) wavelet frame, it exists in all the frames of the considered video scene. Another promising temporal transform is Independent Component Analysis (ICA). This transform produces a set of frames which can be used as independent sources to generate the processed video sequence. The highly semantic role of the extracted components open avenues to produce watermarks which are related with the video scene [SLH04].

Since video content can also be regarded as a three dimensional (3-D) signal, the usage of 3-D transforms is usually motivated by visibility and robustness considerations. For instance, 3-D discrete fourier transform (DFT) can be exploited to obtain an alternative representation of a video sequence [DCOP99]. In this case, mid frequencies, should they be spatial or temporal, are considered for watermark embedding to achieve a trade-off between invisibility and robustness against attacks, such as MPEG compression. 3-D discrete wavelet transform (DWT) [KHMV05] and 3-D Gabor transform [ZWH04] have also been investigated to produce robust video watermarks. Nevertheless, considering video as a 3-D signal may be inaccurate. The three dimensions are indeed not homogeneous - there are two spatial dimensions and one temporal dimension

and should not be treated the same way. However this approach remains pertinent in some very specific cases. For example, in medical imaging different slices of a scanner can be seen as different frames of a video. In this case, the three dimensions are homogeneous and a 3-D transform can be used.

2.3 Methods based on Video Compression Standards

Video files are stored most of the time in a lossy compressed version to spare storage space. Similarly, video is usually streamed across digital distribution networks in a compressed form to cut down bandwidth requirements. Therefore, watermarking methods have been designed to directly embed the watermark into the compressed video stream by exploiting some very specific characteristics of the compression standard, such as MPEG, H.264/AVC, etc. For instance, watermarking in the compressed stream can be seen as a form of video editing in the compressed domain [MC96]. Such editing is not trivial in practice and therefore new issues are raised. The Differential Energy Watermarks (DEW) method was initially designed for still images and has been extended to video by watermarking the I-frames of an MPEG stream [LLB98]. To introduce an energy difference, the block discrete cosine transform (DCT) is computed for each 8×8 block and the DCT coefficients are quantized using the standard JPEG quantization matrix. The obtained coefficients are then separated in two halves and the high frequency energy for each region is computed according to the following equation:

$$E(c, n, Q_{JPEG} = \sum_{b=0}^{n/2-1} \sum_{i \in S(c)} ([\Theta_{i,b}]_{Q_{JPEG}})^2 \text{ with } S(c) = \{i \in \{0, 63\} | (i > c)\} \qquad (2.2)$$

where $\Theta_{i,b}$ is the DCT coefficient with index i in the zigzag order in the $b-$th DCT block, [.] indicates the quantization with quality factor Q_{jpeg} and c is a given cut-off index. The value of the embedded bit is encoded as the sign of the energy difference $D = E_A E_B$ between the two regions A and B. All the energy after the cut-off index c in

either region A or region B is eliminated by setting the corresponding DCT coefficients to zero to obtain the appropriate sign for the difference D. Finally, the inverse block DCT is computed and the shuffling is inversed to obtain the watermarked frame. At the decoder, the energy difference is computed and the embedded bit is determined according to the sign of the difference D. This algorithm has been further improved to adapt the cut-off index c to the frequency content of the considered 8×8 block such that the energy difference D is greater than a given threshold [LL01].

Another key element in video coding is motion estimation/compensation to reduce temporal redundancy. Indeed, successive video frames are highly similar and video coding basically aims at predicting one frame from another using motion prediction to reduce the amount of data to be transmitted. For instance, in the MPEG standard, there is a clear distinction between I-frame which is encoded as still image and P or B-frames which are respectively encoded in reference with one I-frame and two other frames, either I or P, respectively. This results in a sequence of motion vectors which are transmitted to the decoder to perform motion compensation on the other side. It could be interesting to consider those motion vectors as potential candidates to carry a secret watermark. For example, the horizontal component of a motion vector is quantized to an even value if the watermark bit to be hidden is equal to zero and to an odd value otherwise. Similarly, for visibility reasons, one can also choose high magnitude motion vectors for embedding and to modify either the horizontal component or the vertical component of the motion vector according to its angle [ZLZ01].

Alternatively, recent advances in digital watermarking with quantization schemes can also be considered for modifying motion information. In this perspective, motion vectors can be quantized with respect to a square grid or a circular grid or an angular grid [BLD03, BLD04]. Such approaches have been demonstrated to be slightly more robust. But one of the major concern when motion information is modified is fidelity: it is very difficult to predict the perceptual impact of modifying motion vectors. Nev-

ertheless, this issue may be not critical in some applications. For instance, motion information can be modified to perform partial encryption, also referred to as water scrambling [YBD04]. In this context, the goal is to degrade the video quality, but still enabling video content to be perceived by an end-user to give an idea of the original content to trigger an impulsive buying action.

In many video encoders, transform domain coefficients and motion vectors are usually quantized, either with a scalar or a vector quantization. The resulting information is represented with some symbols which are sent to an entropy encoder to obtain the final bitstream. For instance, in the MPEG standard, the quantized DCT coefficients are scanned in a zigzag order and represented with (run, level) tuples. The run is equal to the number of zeros preceding a coefficient and the level is equal to the value of the quantized coefficient. Those tuples are then input to an entropy encoder. In practice, some lookup tables are defined in the MPEG standard to associate a Variable Length Coded (VLC) codeword to each possible tuple. As a result, some researchers have investigated how to directly modify the bitstream *i.e.* the VLC codewords to avoid full compression and decompression which is time consuming. In this perspective, a pioneer work has identified a set of VLCs which can be modified without introducing strong visual artifacts [LLB98]. Even if some variations around this approach have been proposed [LCLF02], the most promising research track is the one which exploit recent works to make conventional VLCs exhibit resynchronization properties upon encountering bit errors [MC02, MC04]. Such VLC are called reversible VLCS (RVLC) and are two-way decodable. The idea is then to use the error recovery power of such RVLCs to design reversible watermarking schemes: binary modifications due to the watermarking process are considered as channel errors and recovered.

Some of the previous works have modified directly the bitstream of compressed video to embed a watermark. Such algorithms are very interesting because of the high achievable embedding rate and the low computational complexity. In the context of an

MPEG video stream, a watermark consisting is embedded in the bitstream by selecting suitable VLCs and forcing the Least Significant Bit (LSB) of their quantized level to be equal to the payload bits [LLB98]]. To ensure that the change of VLC is perceptually invisible and that the size of the MPEG stream does not increase, only a few VLC called label bit carrying VLC are considered for embedding. Those VLCs have the interesting property that if an another VLC exists then it will have the same run length, a level difference of 1, and the same VLC length.

Each one of above approaches has its own pros and cons with respect to complexity, robustness performances, visibility etc. as shown in Table 2.1.

Table 2.1 Pros and cons of different approaches of video watermarking.

Approaches	Pros	Cons	Literature
Image watermarking in video	Inherit from all the results for still images	Computationally intensive	[KDHM99, MKHD99, TDSV$^+$00, DSS98]
Temporal dimension based	Video-driven algorithms which often permit higher robustness	Can be computationally intensive	[HG98, MSB02, SZT98, SLH04, DCOP99, KHMV05, ZWH04]
Compression standard based	Simple algorithms which can be be use in real-time	Watermark inherently tied to the video format	[LLB98, LL01, ZLZ01, BLD03, BLD04, YBD04, LCLF02, MC02, MC04]

H.264/Advanced Video Coding (AVC) is regarded as one of the popular and efficient industry standards for video coding, which is built on the concepts of MPEG-2 and MPEG-4. Section 2.4 describes the state of the art literature for watermarking in H.264/AVC encoded videos. The motivation of the thesis is finally elaborated in the Section 2.5.

2.4 Video Watermarking in H.264/AVC

The watermark can be inserted either in uncompressed (raw) video [HG98] or compressed video [MAAK10, LH07]. Video signals are often stored and transmitted in a

compressed format. Application of uncompressed watermarking techniques for compressed video sequences, however, needs full decoding and re-encoding for embedding or watermark detection. In many applications, complete decoding of video sequences is not very suitable. Consequently, compressed video watermarking [MAAK10, LH07] has gained more attention.

A few compressed domain watermarking algorithms for H.264/AVC video have been reported in the literature. Different compressed domain features have been exploited to embed watermark bits. Such features are quantized DCT coefficients, motion vectors, prediction modes, and quantization tables. The watermark algorithms can be categorized based on frames *i.e.* I-frame, P-frame, and B-frame, where the watermark is embedded.

2.4.1 I-frame based watermarking

In [NM05], Noorkami and Mersereau have proposed a blind and fragile watermarking method to resist self collusion attack by extracting keys. The extraction of keys are however requires complete decoding and re-encoding of the compressed video. A watermark is embedded in the least significant bit (LSB) of the quantized AC coefficients of the macroblocks. The authors have improved their method [NM05] by imposing robustness and proposed a non-blind watermarking algorithm in [NM07], where coefficients are selected using the watson's human visual model [Wat93]. The synchronization error is minimized using location map.

Profrock *et al.* have analyzed the H.264 bitstream to generate a watermark sequence and embedded that watermark for authentication [PRSM05]. In literature [ZH06], Zhang and Ho have proposed an embedding algorithm to represent the authentication information from a preprocessed binary watermark sequence and embedded the watermark by changing the sign of the AC coefficients based on the watermark bit. The watermarking algorithm presented in [ZH06] is improved in [ZHQM07] by focusing on

grayscale patterns with characters for preprocessing the watermark. In [ZB09, ZB10], the watermark is embedded by replacing some bits of the bitstream and extraction is performed by detecting meta data generated during the pre-embedding stage.

Mansouri *et al.* [MAAK10] have proposed a blind method where the watermark bits are embedded by changing nonzero coefficients to zero value that significantly degrades visual quality. Block selection is performed based on luminance intra prediction modes and the number of nonzero coefficients of a macroblock. After re-encoding due to synchronization error, intra prediction modes are changed [MAAK10]. The authors in [MAAK10] have investigated the rate of intra prediction mode changes from 4×4 to 16×16 for blocks with different number of nonzero coefficients (NNZ). Blocks containing a higher NNZ have less tendency to change modes. The authors have claimed in [MAAK10] that embedding in the blocks having higher number of nonzero residual coefficients helps to minimize the synchronization error. In another experiment, the authors have further elaborated that the probability of intra mode alternations after re-encoding, decreases for 4×4 blocks with higher number of nonzero coefficients.

Esen and Alantan [EA11] have proposed a watermarking framework that selects embedding region using forbidden-zone-data-hiding and exploits error correction capability feature of Reed-Solomon codes to make it robust. Xu *et al.* [XWW11] have embedded watermark bits by changing the parity of the sign of the coefficients and mid frequency coefficients are chosen for watermarking.

2.4.2 P-frame based watermarking

Nguyen *et al.* [NTD06] have analyzed the H.264 bitstream and embedded watermark in two LSBs of motion vectors. In [NM08], Noorkami and Mersereau have proposed a variation on the watermark detection algorithm presented in [NM07], where the non-blind watermark is embedded in all nonzero coefficients without using any visual model. The synchronization error is minimized using location map.

Kuo *et al.* [KLL08] have embedded a blind and fragile watermark in the LSB of motion vectors. The LSB of horizontal and vertical motion vectors are XORed to match with watermark bit otherwise either of the LSB of horizontal and vertical motion vectors are replaced with the watermark bit. Swaraja *et al.* [SLRP11] presented an algorithm that divided the search space in even and odd positions and based on that the watermark was embedded in the LSB of motion vectors. Feng and Wu [FW11] have selected 8×8 mode blocks to embed a blind watermark by changing the parity of horizontal and vertical components of motion vectors. In the preprocessing phase, the watermark is extracted based on the energy of macroblocks in I-frames.

2.4.3 Watermarking in I-frame and P-frame

Qiu *et al.* have embedded a non-blind watermark into DCT coefficients of I-frames and motion vectors of P-frames [QMH+04]. Kuo and Lo have improved the technique proposed in [QMH+04] by choosing more appropriate locations for embedding [KL10]. Su *et al.* [SWC+11] have proposed a non-blind method to embed watermark in I-frames and P-frames. A non-blind watermark is embedded using spread spectrum and coefficients are selected using the watsons human visual model [Wat93].

2.4.4 Drift compensated watermarking

In most compressed domain methods, error introduced by watermark embedding propagates into subsequent frames due to intra or inter prediction that degrades the visual quality of the video. In such a scenario, maintaining acceptable visual quality of watermarked video is an important issue. To eliminate the effect of drift error propagation due to embedding in intra predicted macroblocks of I-frames, Gong *et al.* in [GL08] have proposed a non-blind algorithm, which has added a drift compensation signal to the carrier signal. Chen *et al.* in [WCC08] have exploited several transform coefficients of a macroblock for embedding a watermark bit. The algorithm presented

in [GL08, WCC08] requires the original watermark for extraction at the decoder. The algorithm proposed in [WCC08] is improved by Ma *et al.* [MZTZ10] by exploiting several coefficients of 4×4 blocks based on the directions of intra frame prediction modes. Figure 2.1 shows the predicted samples (from a to p) in a 4×4 block in a current block, denoted by $B_{i,j}$, are obtained from the adjacent samples (from A to M) based on the selected prediction mode. The values $\{i, j\}$ represent the i^{th} row and j^{th} column of 4×4 blocks in a I-frame.The drift error is compensated based on the predicted samples (from a to p) in such a way that edge coefficients {d,h,l,m,n,o,p} will remain unchanged.

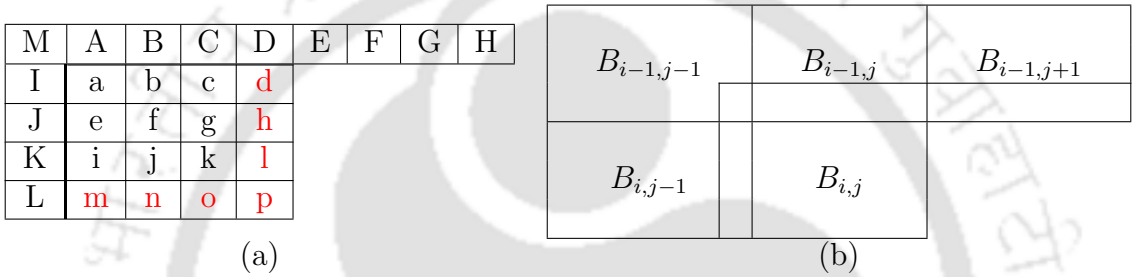


Figure 2.1 The intra predicted samples (from a to p) of a 4×4 block ($B_{i,j}$) and its adjacent samples (from A to M) in a I-frame.

A non-blind drift compensation algorithm proposed in [ZZP10] where the error propagation is eliminated by obtaining the difference between original and watermarked reconstructed samples. Quantized coefficients are compensated according to the corresponding differences. The algorithm [ZZP10] is improved by Huo *et al.* in [HZC11], where quantized coefficients are classified into three different categories for appropriate drift compensation. The algorithms proposed in literature [GL08, WCC08, MZTZ10, ZZP10, HZC11] are to prevent distortion drift in intra predicted blocks for watermark embedding in I-frames.

Xiao *et al.* [XYH+] have proposed a non-blind drift compensation method using the reversible watermarking technique in I-frames and P-frames, where full decoding and re-encoding of the compressed video stream is required for embedding to prevent distortion drift in both intra and inter predicted blocks. Facciol and Farrugia in literature [FF10] have proposed a blind embedding algorithm which adapts reversible

watermarking [Tia03] to avoid propagation of embedding error within every block.

2.4.5 Attack resistant watermarking

A video watermarking algorithm has to be robust; subsequent processing of watermarked data should not impair the detection of embedded information [PB06]. Video signals are susceptible to image and video processing attacks and hostile attacks. Common image processing attacks are the addition of noise like salt and pepper noise, gaussian noise, etc., filtering with circular averaging filter, gaussian filter, etc. The hostile attacks include frame drop or alter attack, re-encoding attack, collusion attack, and rotation-scaling-translation attacks, etc.

Due to synchronization error (refer to section 1.1.4), robustness of an embedding technique decrease, which make easy for attackers to remove the watermark from the watermarked video with little degradation in visual quality. Moreover, the authorized client may receive a weak watermark due to synchronization error, which can create confusion in authenticity. Location map is generally used to minimize synchronization error [NM07, MAAK10, EA11, NM08]. A location map is a file which contains the exact location of blocks or coefficients, where the watermark bits are embedded. Secured transmission of location map is still a potential overhead. The watermark can be completely destroyed if the attacker can able to access the location map.

Mansouri *et al.* have presented a blind compressed domain algorithm to embed in I-frames of H.264/AVC videos [MAAK10]. The synchronization error is minimized based on the luminance intra prediction modes and number of nonzero residuals in a macroblock. However, to the best of our knowledge, no watermark embedding algorithm is still proposed to minimize synchronization error for inter predicted macroblocks.

Vinod *et al.* in [PDB09] have designed a tool to check whether a video sequence contains any motion incoherent component using features extracted from frame prediction error. Video watermarking in a frame-by-frame manner can directly incorporate a well

developed image watermarking techniques [JhG00], but discards an important feature of video, *i.e.*, temporal redundancy across successive video frames. Embedding same watermark in all frames of a video is robust to Type II collusion attack, but vulnerable to Type I collusion attack and embedding uncorrelated watermarks in different frames of a video is robust to Type I collusion attack, but susceptible to Type II collusion attack. The straightforward adaptations of image watermarking methods [JhG00] in a frame-by-frame manner are therefore insecure in terms of collusion attacks. Several attacks have recently been identified to defeat commonly used video watermarking algorithms [LH07, SKH05]. In order to withstand temporal frame averaging along the motion axis, motion coherency has recently been identified as a desirable property for embedding watermarks within video streams [PDB09].

The practical implementations of approaches on collusion attacks rely on rigid frame registration techniques [DD04], temporal transforms using motion compensation [PB06], or some shaping of watermark according to the motion information [HM05]. A collusion attack can be designed based on motion compensated temporal-frame averaging (MC-TFA) [PDB09]. The host estimate U_i' is compared to the given video Y_i in order to detect if something is hidden. The basic hypothesis is that the deviation of specific characteristics of U_i' and Y_i will differ if something is embedded in Y_i, such that,

$$Y_i = U_i + \alpha W_i \tag{2.3}$$

in comparison to when nothing is embedded in Y_i, *i.e.*,

$$Y_i = U_k \tag{2.4}$$

The collusion is done by exploiting the differences and similarities among frames to judiciously reduce the energy of the watermark in relation to that of the host information

[UBZ06]. The collusion of a sequence of video frames is represented as follows

$$U_i' = \S_T(Y_i) = \zeta[Y_1, Y_2 \ldots, Y_n] \tag{2.5}$$

where U_i' is called the colluded result and represents the estimate of the i^{th} host frame U_i. § is the motion compensated collusion operator with parameters Th that exploits the similarities and differences among all or a select subset of possibly watermarked image frames $[Y_1, Y_2 \ldots, Y_n]$ to produce U_i'. Motion compensation is done among the watermarked image frames $[Y_1, Y_2 \ldots, Y_n]$ [PDB09]. A sliding window of size $2T + 1$ is used to denote the temporal neighborhood used for frame averaging and this window is assumed to contain visually similar frames. Let $\kappa_{i \to i+1}(n)$ denote the position of a point in the frame Y_i which has moved to the position in the frame Y_{i+1}. The motion-compensated prediction of the frame by using the reference frame is defined as

$$Y_i^{i+1}[n] \equiv Y_i[\kappa_{i \to i+1}(n)] \approx Y_{i+1}(n), \quad n \in \lambda \tag{2.6}$$

where $\lambda = [1, N_1] \times [1, N_2] \subset \mathbb{N}^2$ is a 2-D grid of points representing the position of the pixels in a video frame of dimension $N_1 \times N_2$.

When a watermarked sequence is temporally averaged with an odd window length equal to $2T + 1$, the resulting attacked frames U_i' are given by

$$U_i' = \S_T(Y_i) = \frac{1}{2T+1} \sum_{j \in W_i} Y_j = \frac{1}{2T+1} \sum_{j \in W_i} U_i + \frac{1}{2T+1} \sum_{j \in W_i} W_i \tag{2.7}$$

where $W_i = \{i - T, i - T + 1, \ldots, i + T\}$ is the set of temporal indexes included in the filtering window.

2.5 Research Motivations and Objectives

In the aforesaid literature, algorithms presented in [KLL08, SLRP11, FW11, NTD06] are based on the LSBs modification or replacement, where the watermark is embedded in the LSBs of motion vectors in P-frames. Such LSB based methods [KLL08, SLRP11, FW11, NTD06] are fragile to common image and video processing attacks. It is observed that the degradation in visual quality and increase of bit rate are higher in [KLL08, SLRP11, FW11, NTD06] algorithms. Moreover, full decompression and re-compression of video for embedding is required in [NTD06]. The original video is required at the decoder for extraction of the watermark and increase in the video bit rate is high in algorithms presented in [NM08, SGD12]. FDAS attacks are not handled in none of the aforesaid P-frame based algorithms. To the best of our knowledge, no P-frame based embedding method exists in literature, which is simultaneously blind and robust in nature. It motivates us to propose a novel P-frame based watermarking method which is not only robust and blind in nature but also provide better visual quality with marginal increase in bit rate of the watermarked video in compressed domain. Necessary measures would be taken to resist FDAS attacks.

Apart from watermarking parameters (such as robustness, perceptual quality, security, etc.), there are two main challenges exist in compressed domain video watermarking. They are 1) distortion drift and 2) synchronization error. Due to perturbing the compressed domain parameters like residual coefficients, other parameters like prediction modes may get changed. Changes in prediction modes due to embedding may affect the synchronization in the watermark extraction process even with same configuration parameters. In [MAAK10], the authors have tried to minimize synchronization error due to watermarking distortion in intra predicted blocks using an efficient block selection process. Another crucial problem of decoder based compressed domain watermarking algorithms is drift error propagation, which refers to the watermark error accumulations among different blocks during intra or inter predictions [MZTZ10]. A

very few attempts are made to propose video watermarking methods that can completely handle drift distortion. Handling the drift error propagation due to P-frame embedding is a challenging issue as different intra and inter prediction modes coexist and motion estimation is performed over multiple reference frames. Manipulating some of the coefficients of the nearby blocks will not serve the purpose. Slight modification in a block may change the block size or even may change the reference frame. Therefore, algorithms proposed in [MZTZ10, HZC11] may not be suitable for this purpose. Use of reversible watermarking may be an option. The algorithms proposed in [FF10, XYH+] have used reversible watermarking to prevent distortion drift. However, the methods [FF10, XYH+] are based on LSB matching methods and thus fragile to common image and video processing attacks. Similarly, the reversible embedding methods [Tia03, CC07, NSAS06, THY09, LYK07] exist in the literature are mainly fragile in nature. In [XYH+], the watermarking method is non-blind in nature and requires complete decoding and re-encoding of the compressed video stream for embedding. To the best of our knowledge, no robust watermarking method that can handle synchronization error and distortion drift in P-frame is reported in the existing literature. It motivates us to propose a robust reversible watermarking method to avoid drift error propagation an efficient watermarking zone selection algorithm for P-frame embedding.

The I-frame based methods [MZTZ10, NM05, EA11, XWW11] found in the literature are fragile in nature. These algorithms are mainly based on LSB matching or replacement technique. These watermark embedding algorithms are not even robust to common image processing attacks. Moreover, the embedding framework [EA11] increases video bit rate significantly and the number of blocks suitable for embedding is relatively less. In [NM07, SWC+11], a computationally expensive prediction process is required for watermark embedding. The complete decompression and re-compression of the video is required. Furthermore, the synchronization error due to embedding is

not handled. This significantly degrades the visual quality and enhances synchronization error. Moreover, the embedding capacity is very low. Drift error propagation is not prevented in [MAAK10, NM07, NM05, EA11, XWW11]. The embedding capacity of [MAAK10] is low because changing number of nonzero coefficients highly increases the synchronization error. As the methods [NM07, HZC11] are non-blind in nature, the original video is required for extraction of watermark at the decoder. The drift compensation performed in literature [MZTZ10] is only for intra predicted blocks. The techniques [MZTZ10, HZC11] is time consuming with higher bit increase rate (BIR). A complex drift signal compensation technique is proposed in [HZC11], which increases the computational complexity. It motivate us to design a blind I-frame watermarking algorithm, which have higher perceptual quality of the watermarked video compared to existing I-frame based watermarking algorithms. The watermarking method would have a limited increase in video bit rate and compensate drift error propagation (alike [MZTZ10]). The watermarking algorithm would be robust in nature (unlike [MZTZ10]).

One of the potential attacks to the video watermarking is the collusion attack. A few literature exists that can resist collusion attacks. The algorithms proposed in [NM07, LH07, KB11] to deal with collusion attacks, however, fails to resist motion compensated-temporal frame averaging (MC-TFA) attack. Budhia *et al.* have applied a block based motion compensation before temporal frame averaging which highlight that the signal of interest is basically the prediction error after motion compensation [UBZ06]. However, the technique is not very efficient to protect from MC-TFA attack. The framework proposed by Vinod *et. al.* [PDB09] is designed to detect the incoherent object in a watermarked video (*i.e.* MC-TFA attack) and provide solution for prevention for uncompressed videos. It is clear that no motion coherent watermarking method, which is robust to MC-TFA attack, is designed in compressed domain. This motivates us to propose a detection method for motion coherent blocks using only compressed domain

features and analyzing its robustness against such collusion attacks.

The main objectives of this dissertation are to enhance the performance of the video watermarking algorithms for both I-frame and P-frame embedding in compressed domain, which can be summarized as follows:

1. Increasing the robustness of the existing methods by handling drift error and reducing synchronization error.

2. Improving visual quality and controlling bit increase rate by carefully selecting watermarking regions and coefficients using spatial and temporal analysis.

3. Resisting collusion attacks using motion coherent watermarking by exploring the compressed domain features.

4. Preventing FDAS attacks using different error correction codes based on the application.

Chapter 3

Robust Watermarking in P-frame

As mentioned in the previous chapter, video watermarking has become an essential tool for copyright protection and content authentication. A video sequence is modeled as a three dimensional signal where temporal motion is also an important aspect. Therefore, direct extension of image watermarking algorithms (assuming that the video sequence as a collection of images) does not provide optimal solution in most of the applications. Videos have a huge amount of data and inter-frame correlation between successive frames are generally very high, so videos are stored and transmitted in a compressed format. Intuitively, compressed domain watermarking methods are computationally less expensive as complete decoding and re-encoding is not necessary for watermark embedding and extraction. The compressed domain watermarking is quite popular for its reduced computational cost. However, watermark embedding in compressed domain makes video watermarking algorithms susceptible to synchronization error, which decreases the robustness of a watermarking algorithm. Moreover, maintaining an acceptable visual quality of the watermarked video is also a very challenging task.

In H.264/AVC based video watermarking literature, the algorithms proposed in [NM07, MAAK10, EA11] have embedded watermark in I-frames as these frames are crucial for

videos. I-frames are intra predicted frames and carry most of the information of a GOP. P-frames and B-frames in a GOP are inter predicted frames and contain only differential frame information. Intuitively, embedding in I-frame is not prone to frame dropping, frame averaging, and frame swapping (FDAS) attacks. However, the embedding error propagates in the remaining uncoded (not yet encoded) blocks of that I-frame due to the intra prediction and P-frames and B-frames due to the inter prediction in that GOP. This significantly degrades the visual quality of watermarked video. B-frames are highly sparse due to bidirectional motion estimation. So, the majority of coefficients is zero. Watermark embedding in zero coefficients significantly increases video bit rate and degrades visual quality. Therefore, the number of suitable blocks for embedding in B-frames is very less.

The embedding capacity of the P-frame is less compare to I-frame because the P-frames are compressed using differential encoding. However, the P-frames occur more frequently than I-frames within a GOP. Since, sufficient number of nonzero coefficients (NNZ) exist in P-frames, the embedding capacity of P-frames can be exploited. Watermarking in P-frames gives better visual quality than I-frames [NM08]. The embedding error propagates only in uncoded P-frames and B-frames in that GOP. Therefore, it can be concluded that embedding watermark in P-frames provides better perceptual quality than I-frames and higher embedding capacity than B-frames. However, P-frame embedding methods may prone to FDAS attack.

In the state of the art literature in Chapter 2, the algorithms presented in [KLL08, SLRP11, FW11, NTD06] are based on the least significant bits modification or replacement, where the watermark is embedded in the least significant bits of motion vectors in P-frames. It is observed that the degradation in visual quality and increase of bit rate are higher in [KLL08, SLRP11, FW11, NTD06] algorithms. Moreover, full decompression and re-compression of video for embedding is required in [NTD06]. Such least significant bits based methods [KLL08, SLRP11, FW11, NTD06] are fragile to common

image and video processing attacks.

In [NM08], Noorkami and Mersereau have proposed a non-blind watermarking algorithm for embedding in all nonzero quantized coefficients P-frames, where the complete decompression and re-compression of the video is required. In [SGD12], a non-blind embedding is done for P-frames and B-frames by modifying the low frequency coefficients. The original video is required at the decoder for extraction of the watermark and increase in the video bit rate is high in algorithms presented in [NM08, SGD12]. FDAS attacks are not handled in none of the aforesaid P-frame based algorithms.

A major challenge for compressed domain video watermarking is to resist synchronization error. Due to synchronization error, robustness of the embedding algorithm decrease, which make easy for attackers to remove the watermark from the watermarked video with little degradation in visual quality. After re-encoding due to synchronization error, intra prediction modes are changed [MAAK10]. The authors in [MAAK10] have investigated that the rate of intra prediction mode changes from 4×4 to 16×16 for blocks with different NNZ. Blocks containing higher NNZ have less tendency to change modes. The authors have claimed in [MAAK10] that embedding in the blocks having higher NNZ helps to minimize the synchronization error. In another experiment, the authors have further elaborated that the probability of intra mode alternations after re-encoding, decreases for 4×4 blocks with higher number of nonzero coefficients.

P-frame embedding methods are more prone to distortions, specially synchronization error. In P-frames, luminance intra and inter prediction modes coexist. Different luminance inter prediction modes are $\{4 \times 4,\ 4 \times 8,\ 8 \times 4,\ 8 \times 8,\ 8 \times 16,\ 16 \times 8,\ 16 \times 16,$ SKIP$\}$. There are 10 different prediction modes in P-frames and prediction modes are changed frequently due to embedding. So, locating the embedding zone (based on prediction mode) might be hard. In P-frames based watermarking algorithms, such as, [KLL08, SLRP11, FW11, NTD06], the synchronization error is not handled. Handling synchronization error due to P-frames embedding is an important motivation

as different intra and inter prediction modes coexist in P-frames.

The rest of the chapter is organized as follows. In the next section, the motivation of this work is described. In Section 3.2, the proposed method is described. The simulation results are shown in Section 3.3 and finally the chapter is concluded in Section 3.4.

3.1 Motivation

P-frame based embedding methods described in [KLL08, SLRP11, FW11, NTD06] are blind and fragile in nature. The methods in [NM08, SGD12] are robust and non-blind in nature. To the best of our knowledge, no P-frame based embedding method exists in literature, which is simultaneously blind and robust in nature. It motivates us to propose a P-frame based watermarking method which is not only robust and blind in nature but provides better visual quality with marginal increase in bit rate of the watermarked video.

In this work, a robust and blind P-frame based watermark embedding method with a controlled increase in bit rate and having a better perceptual quality of watermarked video is presented. Dual private pseudo random key is used to select candidate blocks which increase security of the proposed method. The proposed watermark embedding method can also withstand frame dropping, frame averaging, and frame swapping (FDAS) attack.

3.2 Proposed Method

In this section, the proposed P-frame based watermarking method is presented. A readable watermark is embedded invisibly in nonzero quantized AC coefficients in 4×4 blocks of P-frames. The watermark sequence is bipolar (0,1) in nature.

First, a robust watermark is generated that can withstand FDAS attacks. The appropriate blocks and coefficients are selected. Last, the proposed embedding and

extraction algorithms are described by justifying the threshold selection, security, and complexity overhead of the proposed method. The block diagram of the proposed embedding and extraction methods are depicted in Figure 3.1 and Figure 3.2, respectively.

3.2.1 Watermark Generation

It is observed in the literature that the P-frames based watermarking may be prone to FDAS attack. In this subsection, the watermark generation process is described such that the watermarking method withstands variable FDAS attacks. The repeat-accumulate (RA) codes [Joh09] [refer to Appendix] are used to generate a bipolar watermark. The use of repeat-accumulate code makes the method robust against FDAS attacks and enhances the robustness against other common image/video processing attacks.

The actual length of the random watermark, the number of bits embedded per frame, and the total number of P-frames that are watermarked denoted by L, l, and F, respectively. The actual length of the watermark after applying RA code is qL, where a block of length L is repeated q times and q in [Joh09] is given by

$$q = \frac{l \times F}{L} \tag{3.1}$$

The error correcting capability [refer to Appendix] in any code denoted by H can be written as

$$H = \frac{q}{2} = \frac{\text{number of repetitions}}{2} \tag{3.2}$$

Substituting $q = \frac{l \times F}{L}$ in Eq. (3.2), the value of H can be expressed as

$$H = \frac{1}{2} \times \frac{l \times F}{L} \tag{3.3}$$

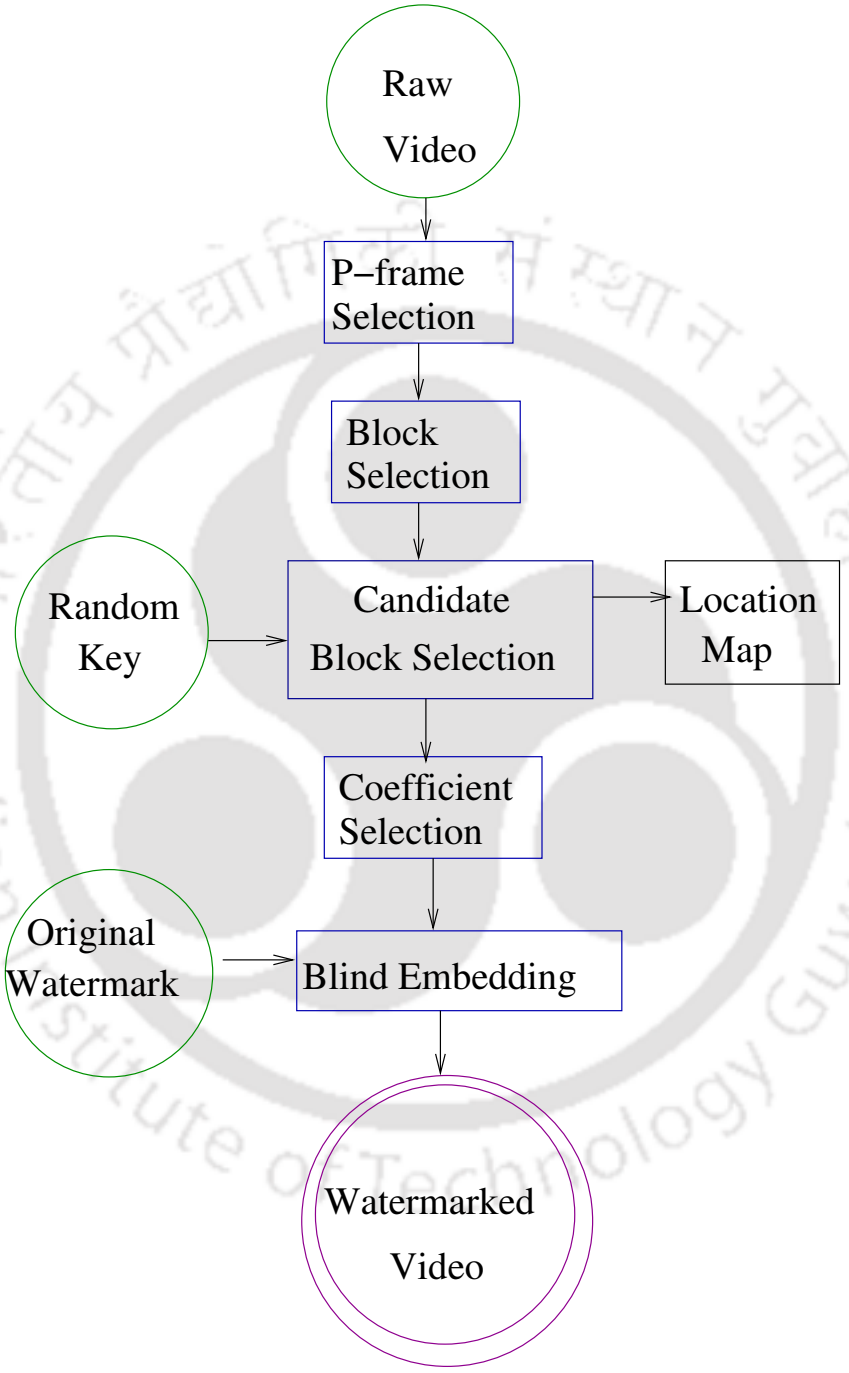


Figure 3.1 Block diagram of the proposed embedding method.

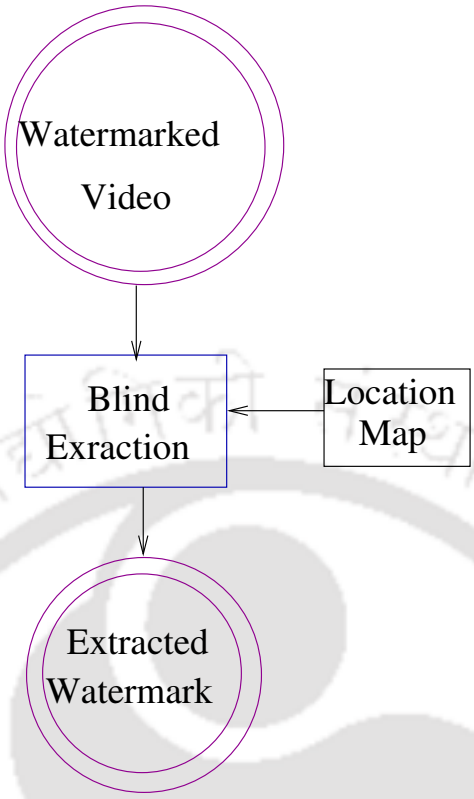


Figure 3.2 Block diagram of the proposed extraction method.

In case of frame dropping, frame averaging, and frame swapping (FDAS) attacks, if the position of a dropped frame or inserted frame can be detected, then erasure channel can be used to enhance error correcting capability of RA code. The error correction rate of FDAS attacks is doubled by the use of erasure channel [refer to Appendix]. Hence, the error correction capability using erasure channel denoted by Z can be written as

$$Z = 2 \times H = \frac{l \times F}{L} \tag{3.4}$$

In case of other image/video processing attacks, the position of the error bit is not known, so use of erasure channel is not suitable. Eq. (3.4) illustrates the relation between error correction capability using erasure channel and ratio of number of bits embedded per frame and length of the watermark. Hence, an optimum value of Z is obtained from the trade-off between the number of bits embedded per frame (l)

and length of the watermark (L) when F is constant, keeping sufficient randomness in watermark generation.

3.2.2 Watermarking zone selection

At first, appropriate blocks are selected for watermark embedding. Then, the appropriate coefficients are selected within a block. In the succeeding subsections, the block selection and coefficient selection processes are described.

Block Selection

The selection of appropriate blocks is very essential to minimize synchronization error and visual artifacts. The appropriate blocks for embedding are selected based on the following parameters: luminance prediction modes, number of nonzero coefficients, and motion vectors. Intuitively, highly textured blocks with motion are considered most suitable for embedding with respect to the fidelity of the watermarked video. Block selection is performed based on following criteria:

- **Number of Nonzero Quantized coefficients:** It has been observed that the number of nonzero quantized coefficients plays an important role in selecting blocks for P-frame embedding that can minimize synchronization error [MAAK10]. Higher the number of nonzero quantized coefficients, less synchronization error is detected. However, the number of nonzero quantized coefficients (NNZ) in intra coded luminance blocks are much higher than inter coded luminance blocks. In P-frames, 4×4 blocks having higher NNZ are very less.

- **Motion Vector Field:** The motion of the block (used for watermark embedding) has a critical role to avoid temporal flickers and artifacts due to embedding distortion. Perturbation in motion regions is more robust against synchronization error. The motion vector in horizontal and vertical direction of a block denoted

by MV_x and MV_y, respectively. All blocks are selected where

$$MV_x \neq 0 \text{ and } MV_y \neq 0 \tag{3.5}$$

Coefficient Selection

After the appropriate blocks have been selected, the compressed domain parameters (such as NNZ, value of quantized coefficients, motion vectors, etc.) associated with these selected blocks are chosen for watermark embedding. To minimize synchronization error, the following parameters are usually avoided for watermark embedding:

- **Number of nonzero quantized coefficients (NNZ):** In [MAAK10], nonzero coefficients are made zero for embedding watermark. Making a nonzero coefficients to zero may change the prediction mode for the future references. It may increase the synchronization error. Moreover, changing nonzero coefficients to zero degrades visual quality significantly. On the other hand, changing the zero coefficients to nonzero may result in a significant increase in video bit rate.

- **Value of motion vector:** In [SLRP11, FW11], motion vectors are changed to embed a fragile watermark. It is observed that changing motion vector values may also affect the future references and thus may lead to increase the synchronization error.

- **DC coefficients or sign of quantized AC coefficients:** Altering DC coefficients or sign of the quantized AC coefficient will degrade the visual quality significantly. This may change in the prediction modes for the future references, which increases synchronization error [NM08].

Based on the above observations, watermark embedding is performed by **changing the absolute value of the nonzero quantized AC coefficients** of each selected

4×4 blocks in P-frames and are considered as a suitable embedding parameter in the proposed work.

3.2.3 Watermark Embedding

The candidate blocks are selected for embedding using a pseudo random key from the set of blocks that selected in block selection (refer to Section 3.2.2). The candidate block selection using a pseudo random key enhances the security of the proposed method. A location map is generated to save the candidate block locations and sent to the decoder. The proposed watermark embedding method for every candidate blocks in each P-frame is described as follows:

1. Nonzero highest coefficient of odd and even sequences of a candidate block in zigzag scan order, denoted by AC_o and AC_e, respectively, where $|AC_o| > 0$ and $|AC_e| > 0$ is estimated as follows:

$$|AC_o| = \text{highest absolute value of } C(n) \text{ for odd values of } n \tag{3.6}$$

   where sign of AC_o is same with that coefficient and $C(n)$ represents the coefficients in a 4×4 block (refer to Fig. 1.12(c)).

$$|AC_e| = \text{highest absolute value of } C(n) \text{ for even values of } n \tag{3.7}$$

   where sign of AC_e is same with that coefficient.

2. The embedding rule is proposed as follows:

$$\textbf{If watermark bit is 0 then } |AC_o| > |AC_e|$$

$$\textbf{If watermark bit is 1 then } |AC_o| < |AC_e| \tag{3.8}$$

3. If watermark bit is zero (0) and $|AC_o| \leq |AC_e|$ then the modified first nonzero coefficient denoted by $|AC'_o|$ is modified as follows:

$$|AC'_o| = |AC_o| + diff_{12} + R_T,$$

where $diff_{12}$ is the absolute difference between $|AC_o|$ and $|AC_e|$ and the robustness threshold R_T ($R_T \in \mathbb{Z}$) is added to increase robustness of the proposed method by keeping significant difference between the coefficients that used for embedding. Thus, at least the magnitude of AC_e need to increase by R_T for incorrect detection of watermark bit, when $|AC_o| = |AC_e|$. Similarly, if watermark bit is unity (1) and $|AC_o| \geq |AC_e|$ then the modified second nonzero coefficient denoted by $|AC'_e|$ is

$$|AC'_e| = |AC_e| + diff_{12} + R_T.$$

If watermark bit is zero (0) and $|AC_o| > |AC_e|$ or watermark bit is unity (1) and $|AC_o| < |AC_e|$, no changes is required. The absolute values of AC_o and AC_e are denoted by $|AC_o|$ and $|AC_e|$, respectively.

The watermark embedding algorithm is given in Algorithm 1. The value of threshold R_T signifies the robustness of the embedding algorithm. It depends on the visual quality. The value of R_T is estimated in Section 3.2.5 based on an exhaustive set of experimental results.

3.2.4 Watermark Extraction

The watermark extraction is performed at the decoder after entropy decoding. The extraction procedure is exactly the reverse process of watermark embedding. The block locations where the watermark is embedded are saved in a location map during the embedding process. This location map is used by the decoder during the extraction

Algorithm 1: Embedding Algorithm

Input: Candidate blocks in P-frames

Output: Watermarked blocks

for each candidate block in P-frames **loop**

Select AC_o and AC_e in odd and even sequences

$diff_{12} = \big||AC_o| - |AC_e|\big|$

if (watermark bit is 0) **and** ($|AC_o| \leq |AC_e|$) **then**

$|AC'_o| = |AC_o| + diff_{12} + R_T$

sign of AC'_o and AC_o are same

$AC'_e = AC_e$

elseif (watermark bit is 1) **and** ($|AC_o| \geq |AC_e|$) **then**

$|AC'_e| = |AC_e| + diff_{12} + R_T$

sign of AC'_e and AC_e are same

$AC'_o = AC_o$

else

no change in coefficients

end

end

process to get candidate blocks. The embedded watermark bit is extracted from every candidate blocks in watermarked video as follows:

$$\text{The watermark bit} = \begin{cases} 0 & |AC'_o| > |AC'_e| \\ 1 & \text{otherwise} \end{cases}$$

where AC'_o and AC'_e are highest coefficients in odd and even sequences of a block in a P-frame of the watermarked video.

The proposed watermarking method is elaborated using two simple illustrative examples.

Example 1 *Assume, the values of two nonzero AC coefficients, denoted by A, B, and R_T, respectively in a block, are 2, 5, and 4, respectively. The absolute difference between A and B will be*

$$|A - B| = 3$$

If watermark bit is 0 then the modified coefficients will be

$$A' = A + |A - B| + t = 9$$

and B remains unchanged. During the extraction process at the decoder, if $A' > B'$ then the watermark bit is 0.

Example 2 *Assume, the values of two nonzero AC coefficients, denoted by A and B, respectively in a block, are 2 and 5, respectively. If watermark bit is 1 then the embedding rule [Eq. (3.8)] is satisfied, so no coefficients will be changed and the value of the watermarked coefficients will be*

$$A = A' = 2 \text{ and } B = B' = 5$$

During the extraction process at the decoder, if $A' < B'$ then the watermark bit is 1.

3.2.5 Threshold Selection

In the proposed watermark embedding algorithm, a threshold R_T is used to increase robustness. The value of R_T is determined based on exhaustive results. The change in visual quality and robustness of the watermarked video are measured using PSNR (using Algorithm 1: Embedding Algorithm) and BER (against recompression error) [refer to Appendix] with the change in R_T is depicted in Table 3.1. If R_T increases, robustness increases, but visual quality degrades. If R_T decreases, robustness decrease, but it results better visual quality. Therefore, a trade-off is considered between visual quality and robustness to select the value of R_T.

Table 3.1 Selection of threshold R_T based on PSNR and Robustness.

Sequence	R_T	Average PSNR	Average Robustness
Carphone	**1**	37.9	80
	4	37.48	81
	7	37.23	83
Foreman	1	36.87	79
	4	36.59	83
	7	36.11	85
News	1	37.77	78
	4	37.48	81
	7	37.30	83
Salesman	1	36.58	79
	4	36.48	82
	7	35.97	83
Suzie	1	37.69	77
	4	37.29	81
	7	36.88	83
Trevor	1	37.04	76
	4	36.71	80
	7	36.49	81

3.2.6 Security

In the proposed method, blocks which are suitable for embedding are selected using block selection method Section 3.2.2. A subset (candidate blocks) among these blocks is again selected randomly using a pseudo random key. In *candidate blocks*, the watermark

sequence is actually embedded. Similarly, a random selection of P-frames is performed using another pseudo random number generator. In other words, the security of the proposed method is imposed by selecting random P-frames and a random subset of previously selected blocks in each selected P-frame using the dual pseudo random key.

Assume, the number of P-frames selected and the total number of P-frames in a video sequence are s and u, respectively. The number of blocks used for embedding is r. The number of candidate blocks selected after spatial and temporal analysis is v. Dual private pseudo random key is used to select s frames out of u P-frames and in each frame r blocks out of v blocks. In the proposed method, the cryptographic space for security is

$$^{u}\mathrm{C}_{s} \times {}^{v}\mathrm{C}_{r}, \qquad u > s, \qquad v > r$$

3.2.7 Complexity and Overhead

The analysis of spatial and temporal characteristics of video streams for selection of blocks and watermark embedding and extraction process using the robust watermarking technique are performed in compressed domain. Therefore, complete decoding and re-encoding of compressed video is not required. This decreases the complexity of the method. Moreover, during the extraction of the watermark, the original video is not required at the decoder. This decreases the complexity of sending original video to authorized clients for extraction of watermark. However, embedding locations (location map) are sent as side information to the decoder. The size of the location map after run-length encoding where information about candidate block locations are saved is negligible compared to the size of the video is shown in Table 4.3.

3.3 Experimental Results

The proposed method is implemented using H.264/AVC [Ric10] reference software JM 17.2 [S08]. In this work, Peak Signal-to-Noise Ratio (PSNR) [refer to Appendix] and

Table 3.2 Average Size of location map Per Watermarked Video

Sequence	Video Size (KB)	location map Size (KB)
Carphone	51	.099
Foreman	67	.102
News	54	.100
Salesman	52	.099
Suzie	55	.100
Trevor	64	.101

Visual Quality Metric (VQM) [refer to Appendix] are used to evaluate the perceptual quality of the watermarked video sequence. The parameters, namely, Bit Increase Rate (BIR) [refer to Appendix] is used to asses bit rate increase due to embedding and Bit Error Rate (BER) [refer to Appendix] is used as robustness metric. Table 3.3 shows the experimental setup.

Intuitively, if R_T is large, robustness will increase but visual quality will degrade. If R_T is small, robustness will decrease but perceptual distortion will be less. So optimizing visual quality and robustness, the value of R_T is taken as 4 after a comprehensive set of trial experimentation (Table 3.1).

Table 3.3 Experimental Setup

Parameters	**Values**
Video Format	Quarter Common Intermediate format (QCIF)
Frame Resolution	176×144 - Horizontal×Vertical
Frame Rate	30 frames per second
Codec Used	H.264/AVC reference software JM 17.2
Intra Period	10
GOP Structure	IBBPBBPBBP
Encoding Profile	High Profile
Entropy Encoding	CAVLC
Number of frames encoded	100 frames per video
Quantization Parameter (QP)	28
Payload	{100, 150, 200, 250, 300}
RA code repeat value (q)	{4, 6, 8, 10, 10}
Threshold R_T	4
Video Sequence	Foreman, Carphone, News, Salesman, Suzie, Trevor

The simulation results for the visual quality of the watermarked video, increase

in video bit rate, and robustness against different attacks are shown in Section 3.3.1, Section 3.3.2, and Section 3.3.3, respectively.

3.3.1 Visual Quality

The visual quality of the watermarked video is evaluated based on PSNR and VQM.

Figure 3.3 and Figure 3.4 illustrate the average PSNR and the average VQM, respectively of the proposed method and compare with the existing P-frame based blind embedding methods [KLL08, SLRP11, FW11] at payload={100, 150, 200, 250, 300} for an average of 100 frames per video. Figure 3.4 depicts that the visual quality of the watermarked video gives acceptable visual quality. It is clear from the result (Figure 3.3) that the proposed method outperforms other methods.

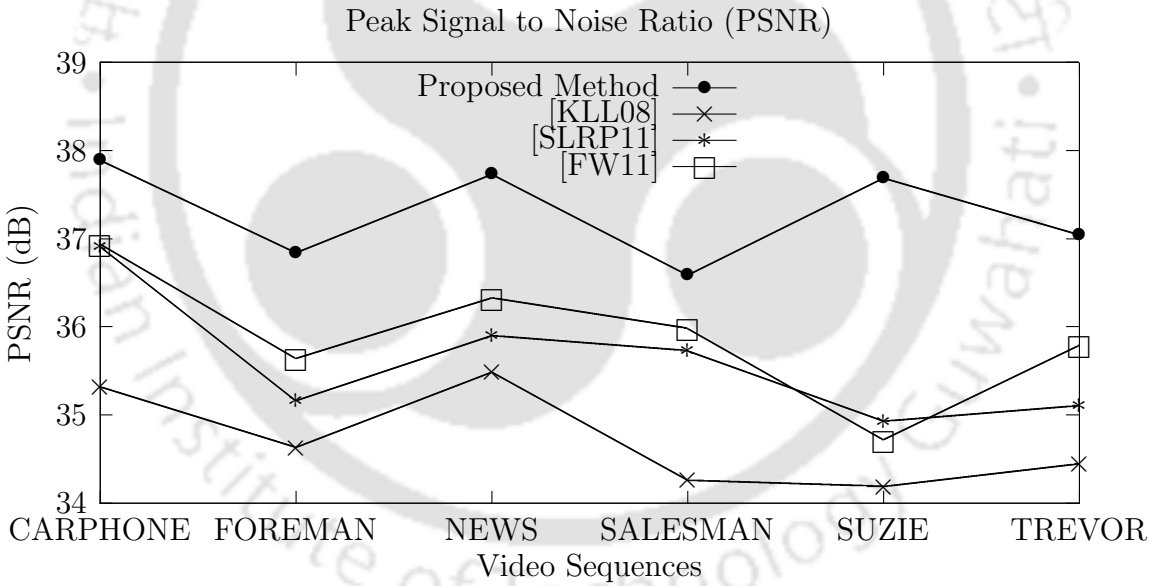


Figure 3.3 The comparison of average PSNR of the proposed method with methods [KLL08, SLRP11, FW11].

3.3.2 Bit Increase Rate

Increase in video bit rate is also calculated and compared with the state of the art literature [KLL08, SLRP11, FW11]. Figure 3.5 shows the average BIR of the proposed

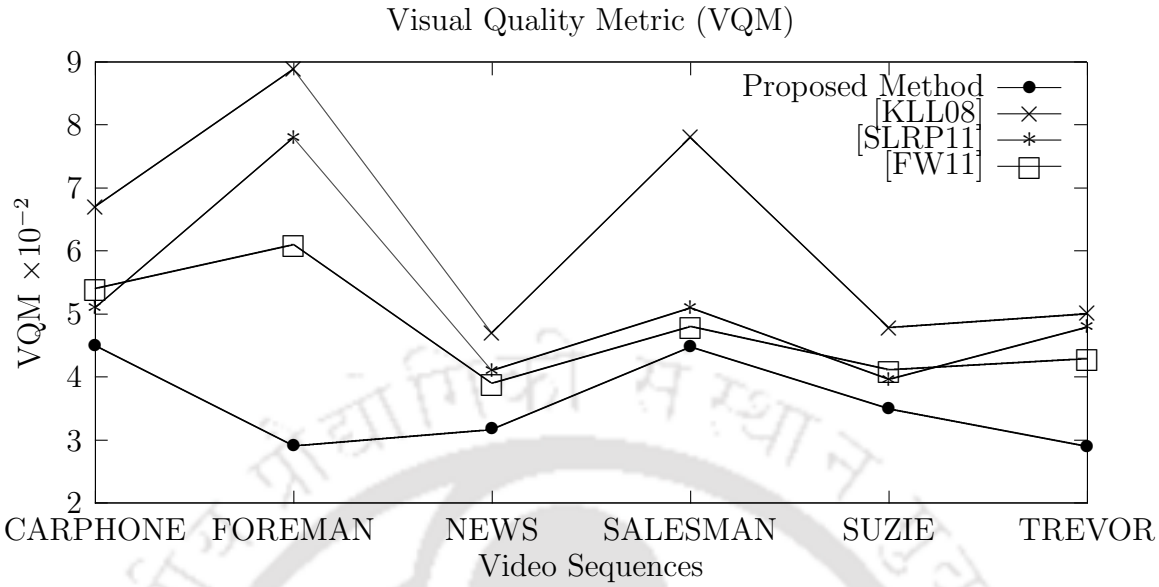


Figure 3.4 The comparison of average VQM of the proposed method with methods [KLL08, SLRP11, FW11].

method and compares with the existing P-frame based blind embedding methods in the literature [KLL08, SLRP11, FW11] at payload={100, 150, 200, 250, 300} for an average of 100 frames per video. The result for BIR in Figure 3.5 shows that the increase in the video bit rate is insignificant and in the order of 10^{-3}.

In the proposed method, only one nonzero AC coefficient in a block is perturbed to embed the watermark. Changing the number of nonzero coefficients (NNZ) or motion vector of a block will significantly degrade visual quality and increase the video bit rate. But in the proposed method, neither of them are changed. Moreover, no zero coefficient is changed to nonzero one or nonzero coefficient is changed to zero one. Hence, in the proposed method increase in the video bit rate is marginal and is in acceptable limit. Intuitively, this is the reason that proposed method outperforms state of the art literature [KLL08, SLRP11, FW11].

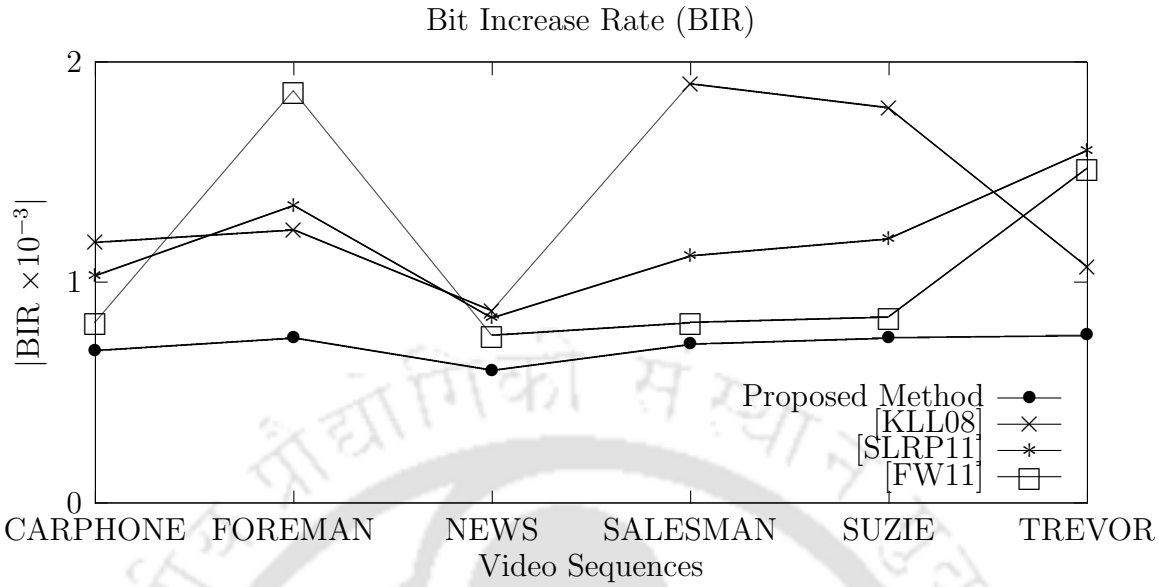


Figure 3.5 The comparison of average BIR of the proposed method with methods [KLL08, SLRP11, FW11].

3.3.3 Robustness to Attacks

In this section, the bit error rate is used for evaluating the robustness [refer to Appendix] of the proposed method against different attacks. The information about the embedding locations is saved as a location map during the embedding process. This location map is provided to the decoder for the extraction process. In the absence of location map, if a watermarked location is not detected or a non-watermarked position is selected incorrectly, the synchronization in the watermark sequence may be lost. This will decrease the robustness of the watermarking method. Intuitively, for a given payload watermark embedding in blocks with higher NNZ increases imperceptibility. In Figure 3.6, it is shown that how NNZ is changing with the alteration of the quantization parameter (QP). It is observed that the NNZ is strictly decreasing with the increase in the change of QP. In the other part of Figure 3.6, it is also observed that the robustness of the watermarking method for changing QP from 28 to range of 20 to 36 in the *foreman* video randomly varies with the change in QP. To get higher compression efficiency with acceptable perceptual quality, the value of QP is chosen

as 28 for the experimentations. Figure 3.7 illustrates the PSNR and robustness of the proposed method in presence of salt and pepper noise, gaussian noise, gaussian filter, and circular averaging filter for the *foreman* video. The PSNR value is scaled in the range of (0,1).

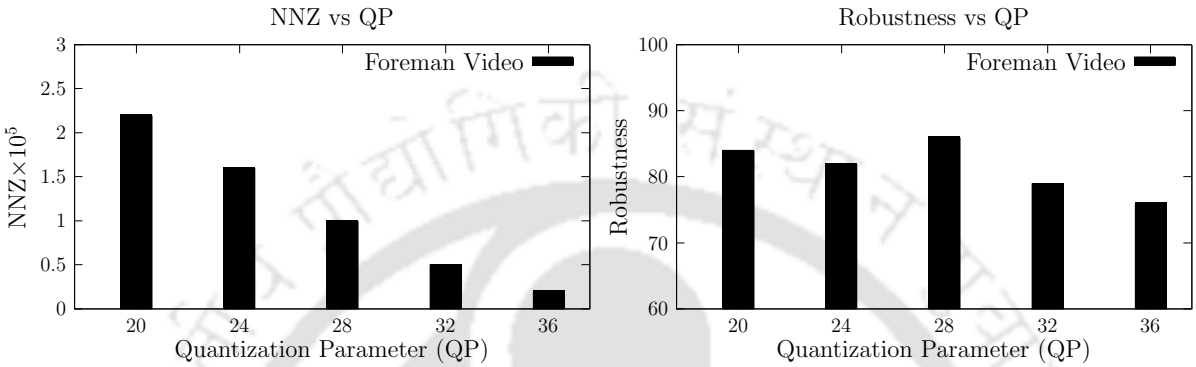


Figure 3.6 The change in NNZ and robustness with QP in the foreman video.

In the Figure 3.8, the comparison of the average robustness against recompression error is depicted. It is observed that the all three variants of the proposed method (with and without block selection, without block selection, and with repeat accumulate coding framework) have outperformed the existing schemes [KLL08, SLRP11, FW11]. Similar results for changing different QP values are depicted in Figure 3.9 and Figure 3.10. In all cases, it is observed that the proposed method is performing relatively better than the existing methods.

The robustness of the proposed blind method (using location map at the decoder) are estimated, using location map at the decoder, against different attacks and compared with the recent literature [KLL08, SLRP11, FW11] against different image and signal processing attacks using location map for extraction of watermark. In the Figure 3.11, the comparison of the average robustness against salt and pepper noise addition (noise density = 0.001) attack is depicted. It is observed that the all three variants of the proposed method (with block selection, without block selection and proposed method with repeat accumulate coding framework) have outperformed the

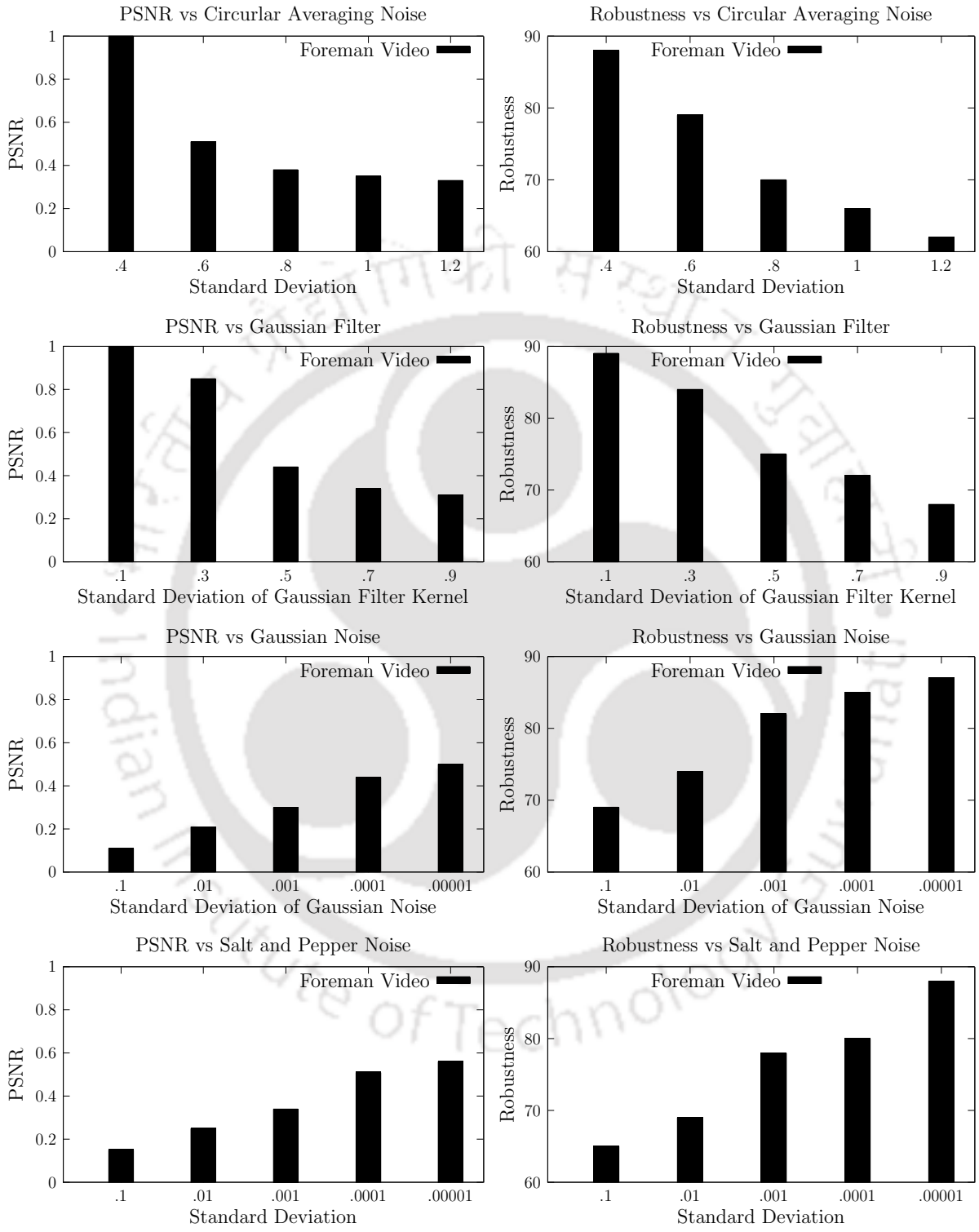


Figure 3.7 The change in PSNR and robustness against different attacks in the foreman video.

existing methods [KLL08, SLRP11, FW11]. Similar results for circular averaging filtering (r=0.06), gaussian filtering, gaussian noise (gaussian noise density = 0.001 and gaussian filter = [5x5], and sigma = 0.3) are depicted in Figure 3.12, Figure 3.13, and Figure 3.14, respectively. In all cases, it is observed that proposed method is performing relatively better then the existing methods [KLL08, SLRP11, FW11]. The locations of these attacks are not known at the decoder so the use of erasure channel may not be beneficial.

In the proposed method, the selection of appropriate blocks helps to minimize synchronization error, which increases the overall robustness of the proposed method. Moreover, no zero coefficients are changed to nonzero or nonzero coefficients to zero nor motion vector is changed. This also helps to reduce synchronization error. Furthermore, only one coefficient in a 4×4 block is modified to embed a watermark bit. All these are intuitive reasons for better robustness of the proposed method in comparison with the state of the art literature.

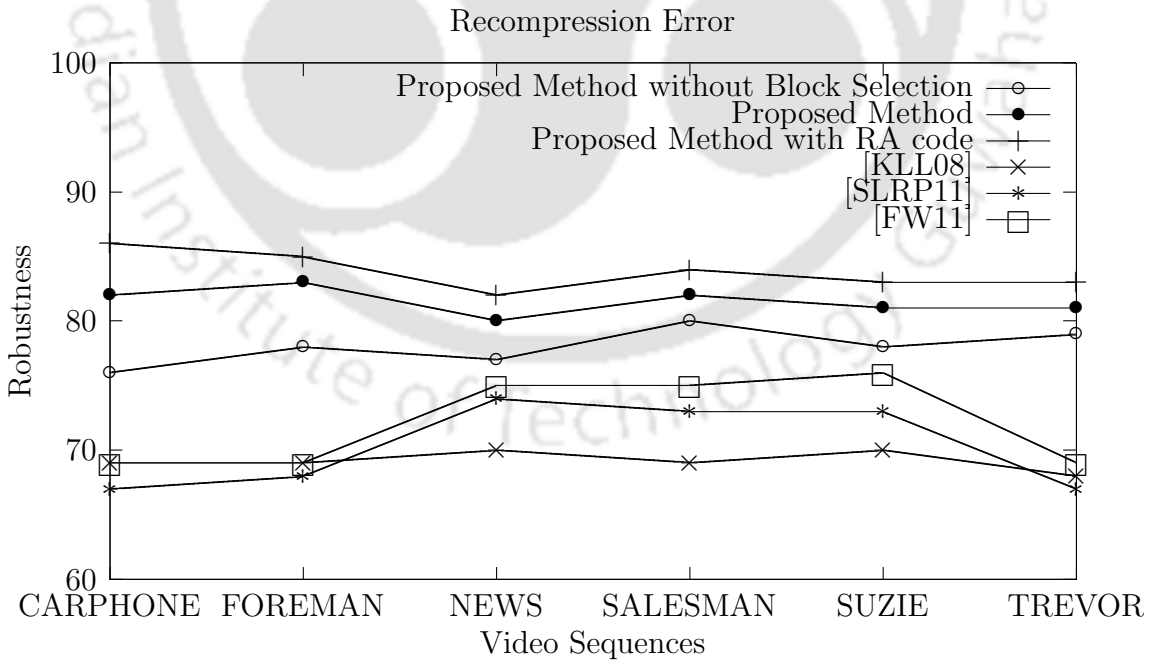


Figure 3.8 The comparison of average robustness of the proposed method against recompression error with [KLL08, SLRP11, FW11].

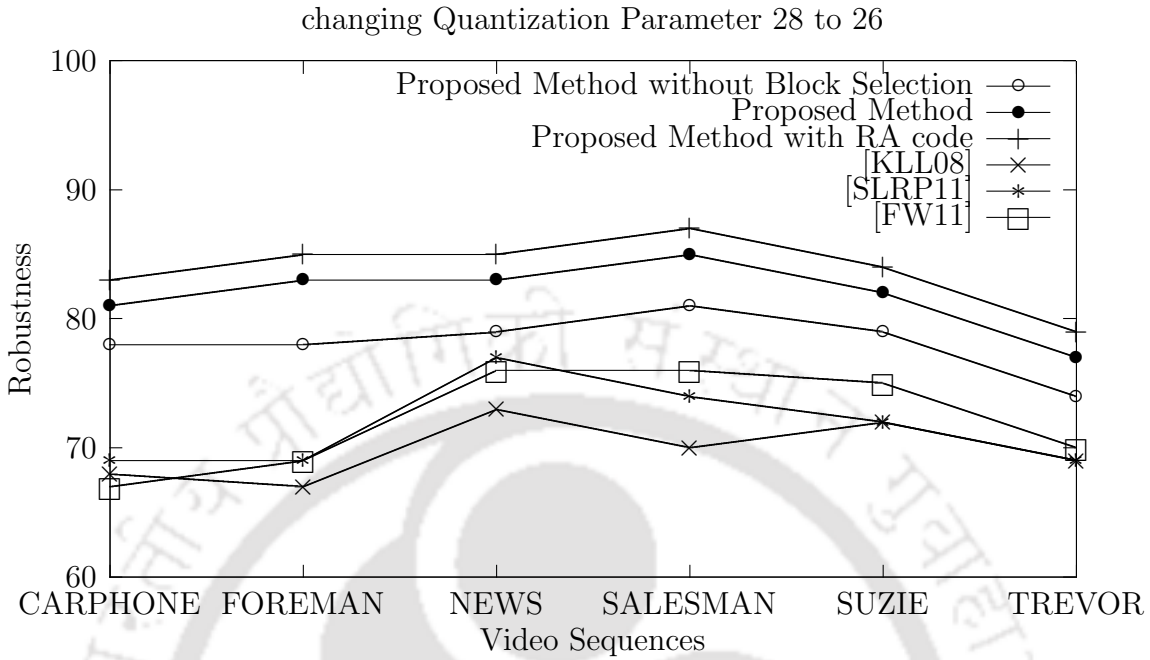


Figure 3.9 The comparison of average robustness of the proposed method against changing QP 28 to QP 26 with [KLL08, SLRP11, FW11].

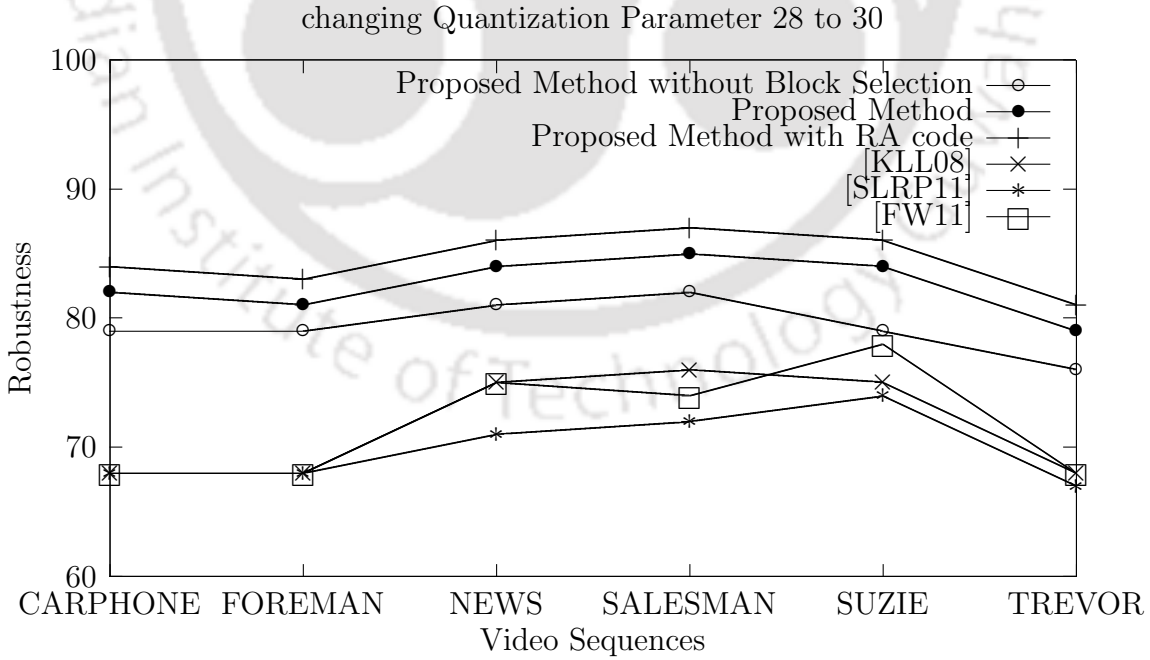


Figure 3.10 The comparison of average robustness of the proposed method against changing QP 28 to QP 30 with [KLL08, SLRP11, FW11].

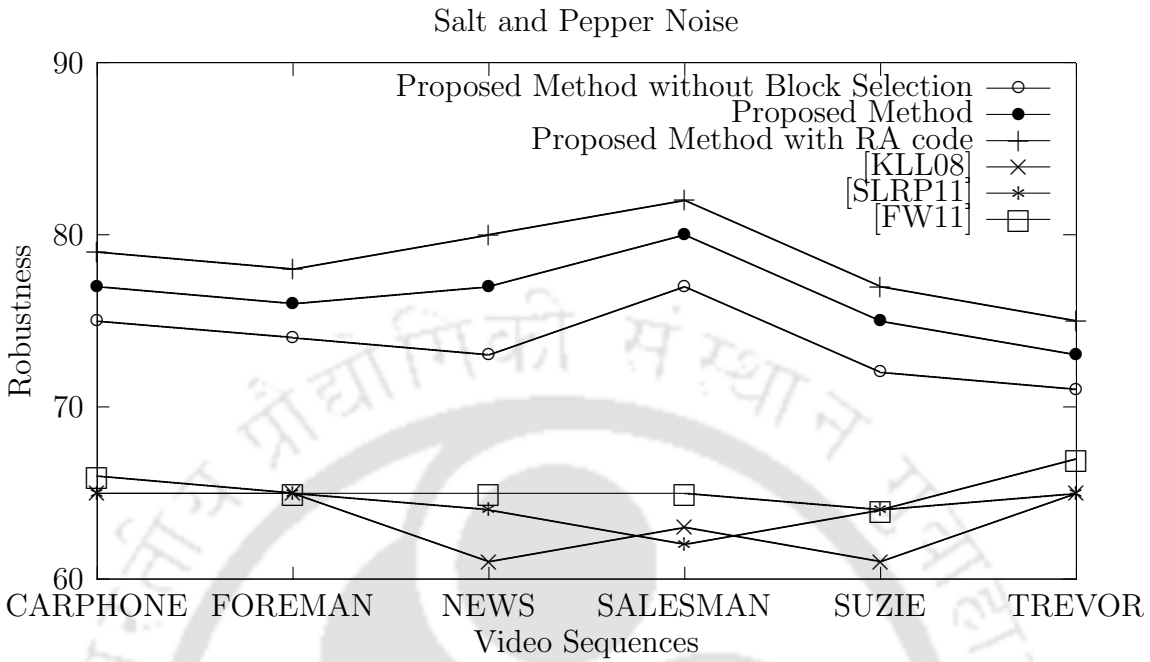


Figure 3.11 The comparison of average robustness of the proposed method against salt and pepper noise with [KLL08, SLRP11, FW11].

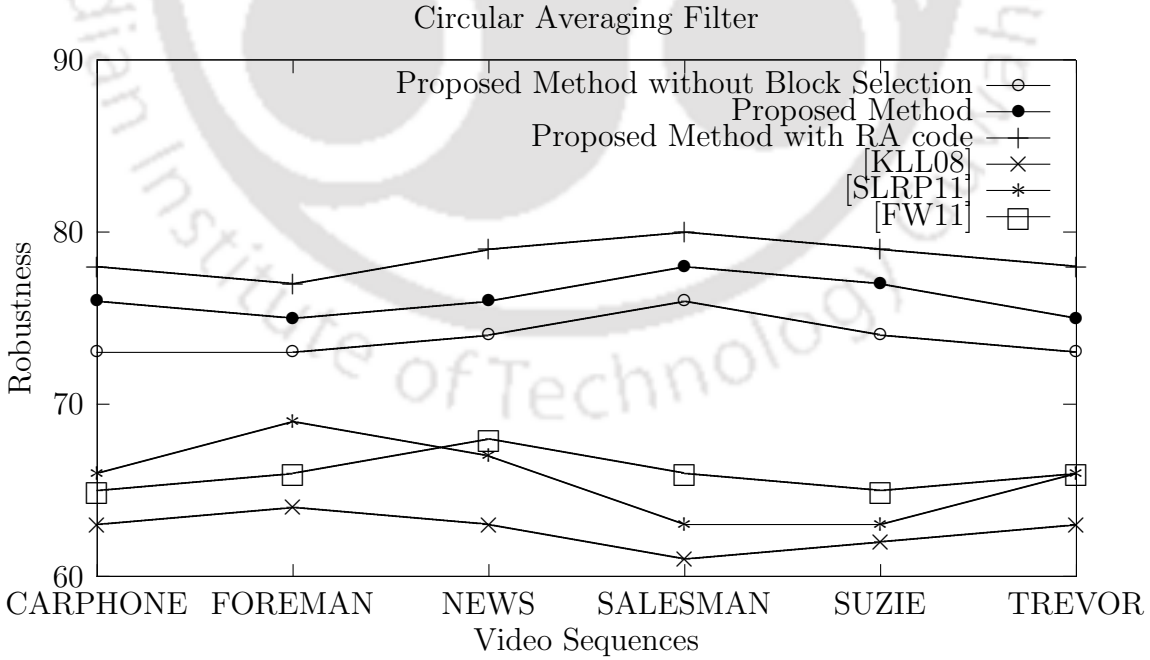


Figure 3.12 The comparison of average robustness of the proposed method against circular averaging filter with [KLL08, SLRP11, FW11].

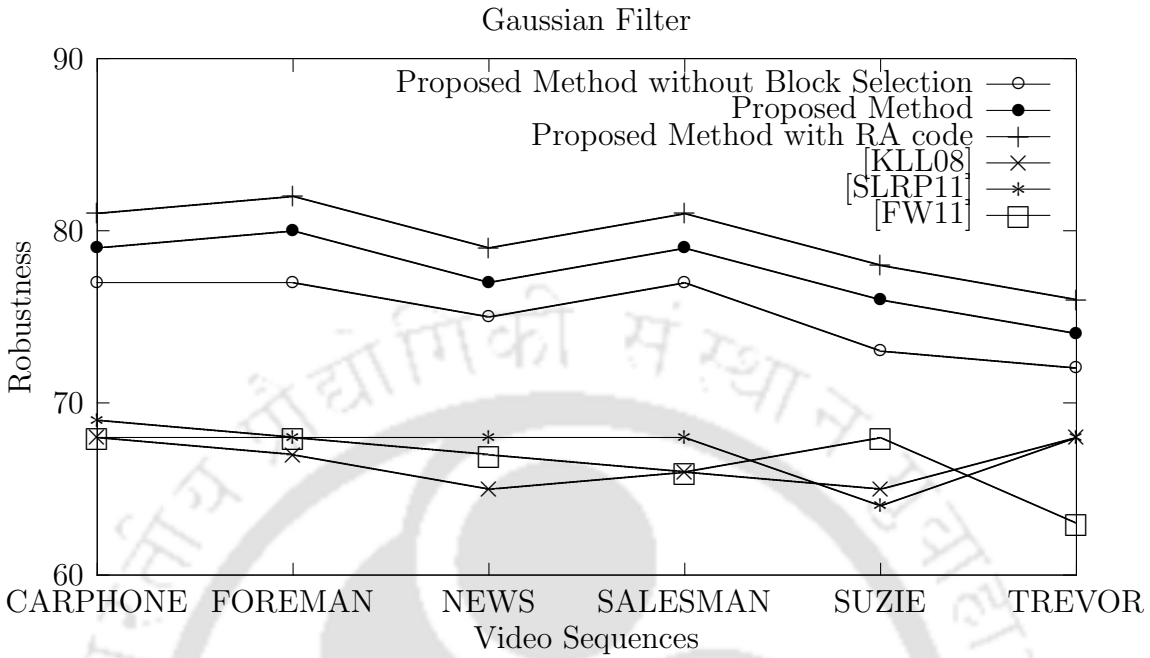


Figure 3.13 The comparison of average robustness of the proposed method against gaussian filter with [KLL08, SLRP11, FW11].

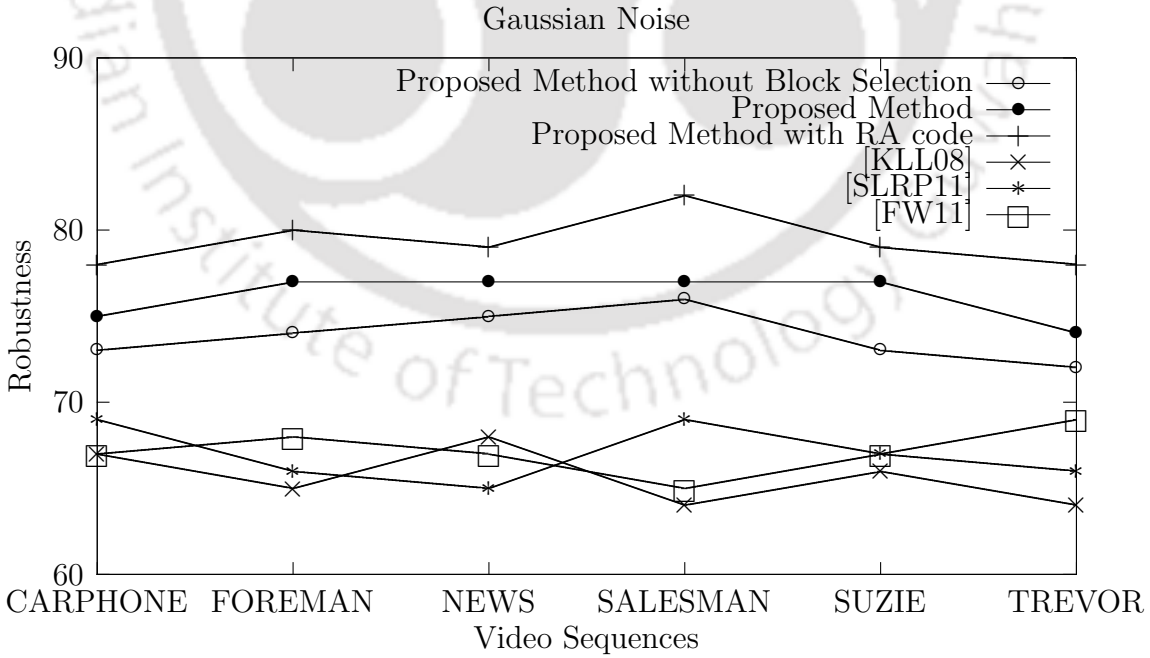


Figure 3.14 The comparison of average robustness of the proposed method against gaussian noise with [KLL08, SLRP11, FW11].

The use of repeat-accumulate codes with an erasure channel in the generation of watermark helps to withstand frame dropping, frame averaging, and frame swapping (FDAS) attacks. The number of P-frames in a sequence of hundred frames of a video is thirty, based on the structure of GOP shown in Table 3.3. The number of bits embedded per frame is unity when payload is hundred and repeat value q is 4, approximately. Hence from Eq. (3.4), four P-frames drop, average or swap can be recovered. The value of q is considered as a small positive integer ($q > 0$). If the value of q is large, then more number of dropping or inserting of P-frames can be recovered, but randomness in the watermark generation will decrease.

3.4 Summary

In this chapter, a P-frame based compressed domain watermarking method, which uses the blind extraction process is proposed. The watermark embedding method has controlled increase in video bit rate and achieve higher perceptual quality of the watermarked video. The proposed method is robust to different image processing attacks. The security is imposed using random keys. Simulation results show the effectiveness of the proposed method with respect to visual quality and increase in video bit rate and compared with existing literature. The result of this work appears in [DSN13a].

Propagation of drift error is a potential problem in compressed domain watermarking that degrades the visual quality of the watermarked video significantly. If the watermark is embedded in P-frames, resisting drift error propagation becomes more challenging. In the next chapter, an watermarking method is designed with the key motivation to compensate the drift error propagation in P-frames using reversible watermarking.

Chapter 4

Drift Compensated Watermarking in P-frame

A crucial problem in the decoder based compressed domain watermarking methods is to handle distortion drift [MZTZ10]. In most compressed domain methods, error introduced by watermark embedding propagates into subsequent frames due to intra prediction in the uncoded blocks of the I-frame or inter prediction in P and B-frames in that GOP, which degrades the visual quality of the video. In other words, it refers to the watermark error accumulations among different blocks during intra or inter predictions. In such a scenario, maintaining acceptable visual quality of watermarked video is an important issue.

To eliminate the effects of distortion drift in I-frames, Ma *et al.* in [MZTZ10] have proposed a blind and fragile watermarking method, where the relationship between the DCT coefficients and the distortion of the pixel values used in intra frame prediction are analyzed to obtain several paired-coefficients as described in Section 2.4.4. One of the coefficients in the paired-coefficient is used for embedding and the other is compensated to fix the distortion on few pixels of the 4×4 block. The algorithm chooses the blocks for embedding according to the intra frame prediction modes of their adjacent blocks

to make sure that the distortion will not propagate to its neighboring blocks.

Huo *et al.* have proposed a drift compensation algorithm by estimating a compensation signal for each block before embedding the watermark sequence [HZC11]. A distortion drift error elimination method with three compensation algorithms for watermarking in H.264/AVC stream is developed. It is indicated that the propagating error is caused by different DCT coefficients, and only some of the DCT coefficients need to be compensated in order to reduce the computational complexity. The difference between original and watermarked samples is computed by compensating quantized residuals. In none of the above methods, the drift error propagation in P-frames and B-frames due to I-frame embedding is not handled.

Xiao *et al.* probably have first introduced the concept of reversible watermarking for handling distortion drift in video watermarking [XYH$^+$] . In the method [XYH$^+$], the authors have proposed a non-blind drift compensation method using the reversible watermarking technique in I-frames and P-frames, where full decoding and re-encoding of the compressed video stream is required for embedding. Facciol and Farrugia in the literature [FF10] have proposed a blind method which adapts reversible watermarking [Tia03] to avoid propagation of embedding error within every block. The algorithms proposed in [XYH$^+$, FF10] are based on least significant bit (LSB) matching methods and thus fragile to common image and video processing attacks. Different techniques are proposed in the literature for reversible watermarking, such as, reversible contrast mapping [CC07], transform domain reversible watermarking [LYK07], different expansion [Tia03], and histogram modification [NSAS06, THY09]. The reversible watermarking technique is generally used for lossless media communication, *e.g.*, watermarking in military, medical and legal imaging, where cover pixels or coefficients that are altered during embedding, can be restored at the time of watermark extraction. The state-of-art reversible watermarking techniques are fragile in nature as these techniques are mostly based on the LSB matching or replacement methods.

The rest of the chapter is organized as follows. In the next section, the motivation of this work is described. In Section 4.2, the framework required for the proposed watermarking method is presented. Next, the proposed method is described in Section 4.3. Then, different features of the proposed watermarking method and the simulation results are illustrated in Section 4.4 and Section 4.5, respectively. Finally, the chapter is concluded in Section 4.6.

4.1 Motivation

From the above discussion, it seems that most of the methods which are proposed in the literature to handle distortion drift are mainly for I-frame based watermarking methods. Handling drift error propagation due to P-frames embedding is a challenging issue as different intra and inter prediction modes coexist and motion estimation is performed over five (maximum) reference frames. Manipulating some of the coefficients of nearby blocks may not be sufficient for the purpose. Slight modification in a block may change the block size or even may change the reference frame. Therefore algorithms [MZTZ10, HZC11] may not be sufficiently robust. Use of reversible watermarking may serve the purpose. But, the existing reversible embedding methods [Tia03, CC07, NSAS06, THY09, LYK07] exists are mainly fragile in nature. To the best of our knowledge, no robust watermarking method that can handle distortion drift in P-frame is reported in the literature.

This motivates us to propose a robust reversible method to handle distortion drift due to P-frame based watermarking. The method would also be blind in nature and have a control in the increase in video bit rate.

In this chapter, a robust reversible watermarking method with blind extraction for H.264/AVC compressed video is proposed which can handle distortion drift. However, a location map describing the embedding positions is sent to the decoder as side information for extraction. The proposed method embeds invisible watermark in P-frames

using a robust reversible watermarking method to prevent distortion drift. Embedding locations are selected by analyzing the spatial and temporal characteristics of a compressed video to enhance perceptual quality and robustness of watermarking method. Embedding is done in nonzero coefficients to restrict increases in video bit rate. It is experimentally shown that the proposed method outperforms the state of the art literature with respect to robustness against re-encoding, re-compression, and other common image and video processing attacks.

4.2 Framework

The efficiency of a video watermarking method is evaluated with parameters like robustness, visual quality, bit increase rate, blindness, and security as discussed in Section 1.1.2. The selection of suitable blocks for embedding makes the watermarking method robust and imperceptible. In a watermarked video stream, two visual artifacts may occur. These artifacts are spatial noise and temporal flicker [WGE03]. Heuristically, low motion (not zero motion) and highly textured areas are suitable for embedding since the human visual system is less sensitive to these areas. In other words, smooth background areas, static or zero motion areas, and high motion areas are heuristically more sensitive to the human visual system so addition of any noise to these areas are easily tracked by human eyes. In addition, different thresholds can be incorporated to enhance the robustness and visual quality of the watermarked video. Therefore, selection of suitable blocks for embedding is an important part for efficient video watermarking.

The spatial and temporal analysis for suitable region selection to preserve robustness and quality, further selection of blocks using watermarking thresholds, and an efficient compressed domain reversible watermarking technique for handling distortion drift are discussed in Section 4.2.1, Section 4.2.2, Section 4.2.3, and Section 4.2.4, respectively.

4.2.1 Embedding Region Selection based on Temporal Characteristics

The analysis of temporal features of a compressed video is an important study for selection of blocks for embedding. Motion information is required to prevent the temporal flicker. Motion analysis is performed based on the available information in compressed domain to avoid further decoding and re-encoding. In this chapter, absolute motion vectors are used to analyze motion information of H.264/AVC video stream. Intuitively, the blocks having low motion (not zero motion) are suitable for embedding as such blocks are less sensitive to the human visual system. Moreover, resynchronization error due to compressed domain embedding will be less for regions having heterogeneous motion [MAAK10]. Therefore, selecting such regions could provide better visual quality and will reduce resynchronization errors.

The block having motion vector equal to zero implies that the block has no motion, which may be due to SKIP (DIRECT) mode of the block. There are no transmitted coefficients in SKIP blocks. Therefore, zero motion blocks are not considered for watermark embedding. Moreover, any perturbation in zero motion blocks and high motion blocks are very sensitive to human eyes. Accordingly, in the proposed work, embedding is restricted to the blocks with motion vector value between one and a given threshold [say *temporal threshold*, (MV_{th}), where MV_{th} is a positive integer]. The process of selecting blocks by analyzing motion vector information is performed in two steps as follows:

Step 1: Calculate absolute motion vector for each block in the P-frame. The motion vector in horizontal and vertical direction of a block denoted by MV_x and MV_y, respectively. The absolute motion vector is estimated based on MV_x and MV_y, which is expressed as

$$\text{Absolute Motion Vector} = \left| \frac{MV_x + MV_y}{2} \right|.$$

Step 2: Select blocks where the temporal mask ($MASK_T$) is one, such that,

$$MASK_T = \begin{cases} 1 & 0 < \text{Absolute Motion Vector } \leq MV_{th} \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

4.2.2 Embedding Region Selection based on Spatial Characteristics

The texture information is another important parameter that describes the repetitiveness of patterns in a frame. Intuitively, perturbing in highly textured areas is less sensitive to the human visual system than perturbing in smooth areas. Highly textured areas are considered as busy areas in a frame [MAAK10]. Moreover, embedding in busy areas helps to avoid the resynchronization error since more textured blocks are often encoded similarly [MAAK10]. So, highly textured regions are preferred for embedding watermark to minimize visual artifacts and the resynchronization error. Furthermore, blocks with higher spatial activity are more robust against different malicious attacks [MAAK10]. However in compressed video stream, blocks in P-frames are sparse due to both inter and intra prediction. Therefore, it is difficult to get texture information from such sparse data. In the proposed method, the texture measure is calculated based on the number of nonzero coefficients (NNZ) of a block in a P-frame. The process of selecting blocks by analyzing spatial information is given in following steps:

Step 1: Select blocks where NNZ is greater than a threshold [say NNZ threshold (NNZ_{th})], such that,

$$\text{Number of Nonzero Quantized Coefficients (NNZ)} > NNZ_{th}$$

Step 2: Select blocks whose spatial mask ($MASK_S$) is one, *i.e.*,

$$MASK_S = \begin{cases} 1 & \text{Number of Nonzero Quantized } AC \text{ Coefficients} > NNZ_{th} \\ 0 & \text{otherwise,} \end{cases} \tag{4.2}$$

where NNZ_{th} are positive integers.

4.2.3 Embedding Region Selection based on Watermarking Thresholds

The blocks are selected for embedding by analyzing spatial and temporal characteristics of the video stream. In addition, to maintain acceptable visual quality and robustness, two watermarking thresholds, namely, visual quality threshold (V_T) and robustness threshold (R_T) are introduced. Visual quality threshold is a parameter by which degradation of visual quality of watermarked video is controlled. Robustness threshold is a parameter, which is used to control the robustness of the proposed method. V_T and R_T are positive integers. The blocks suitable for embedding are selected using spatial and temporal masks ($MASK_T = 1$ and $MASK_S = 1$ of Section 4.2.1 and Section 4.2.2, respectively) along with watermarking thresholds (V_T and R_T of Section 4.3.1).

The coefficients in odd sequence and even sequence are {AC(0,1), AC(2,0), AC(0,2), AC(1,2), AC(3,0), AC(2,2), AC(2,3), AC(3,3)} and {AC(1,0), AC(1,1), AC(0,3), AC(2,1), AC(3,1), AC(1,3), AC(3,2)}, respectively as shown in Figure 1.12(c). The sum of absolute values of coefficients in odd sequence and even sequence, denoted by Odd_{Sum} and $Even_{Sum}$, respectively for a selected block is estimated as follows:

$$Odd_{Sum} = \sum_{\text{n is odd}} |C(n)| \tag{4.3}$$

and

$$Even_{sum} = \sum_{\text{n is even}} |C(n)|, \tag{4.4}$$

where n is the index in Figure 1.12(b). The absolute difference between the sum of the

coefficients in even and odd sequences in a block, where $MASK_T = 1$ and $MASK_S = 1$, is denoted by $diff_{eo}$. is expressed as follows:

$$diff_{eo} = |Odd_{sum} - Even_{Sum}| \tag{4.5}$$

To enhance the robustness of the proposed method and perceptual quality of the watermarked video, robustness threshold (R_T) and visual quality threshold (V_T) are checked, such that,

$$R_T < diff_{eo} < V_T \tag{4.6}$$

Figure 4.1 The blocks, which are suitable for watermark embedding in the seventh frame, i.e., second P-frame and tenth frame, i.e., third P-frame in a GOP of the foreman video.

A block is not used for the embedding if Eq. (4.6) is not satisfied for that block. The blocks selected in the *foreman* video, which are suitable for watermark embedding, are shown in Fig. 4.1. It is observed in Fig. 4.1 that the blocks selected for embedding are mostly on edges and low motion regions.

4.2.4 Drift Compensation using Reversible Watermarking

It is observed in the previous subsection that the distortion due to error propagation is one of the major issues for embedding in the compressed domain. Prevention of error

propagation may be one of the possible solutions to restrict distortion drift. This can be achieved by restoring the embedding coefficients to its original value during extraction of watermark at the decoder. It ensures that prediction of the next frame is made from original coefficients rather than embedded or noisy ones. Thus, no propagation of errors ensures no distortion drift. In this work, restoration of watermarked coefficients to its original value is achieved using a reversible watermarking method. It is noted that reversible embedding is used only to tackle distortion drift problem.

It is observed in the literature that the existing reversible watermarking techniques are mostly fragile in nature ([Tia03, CC07, LYK07, NSAS06]). The video compression process is inherently lossy in nature. The reversible watermarking technique used for embedding must be robust in nature to withstand lossy compression. In this work, a robust reversible watermarking method is proposed as follows:

Reversible Embedding

A bipolar watermark sequence (0,1) is used for embedding. The proposed reversible embedding method is described as follows:

Step 1: Select first two coefficients in a block denoted by A and B, *i.e.*,

$$A \neq 0 \quad B \neq 0 \tag{4.7}$$

The proposed embedding rule is given as follows:

If (Watermark Bit is 0) **then** $|A|$ should be greater than $|B|(|A| > |B|)$

If (watermark bit is 1) **then** $|A|$ should be less than $|B|(|A| < |B|)$ (4.8)

Step 2: If watermark bit is zero (0) and $|A| < |B|$ then the modified first coefficient denoted by $|A'|$ is $D + |B| + 1$, keeping signs of both of them unchanged, where D is the absolute difference ($||A| - |B||$). If watermark bit is one (1) and $|A| > |B|$ then the modified second coefficient denoted by $|B'|$ is $D + |A| + 1$, keeping signs of both of them unchanged. If watermark bit is zero (0) and $|A| < |B|$ or if watermark bit is one (1) and $|A| > |B|$, then no change is required. A location map is generated based on the embedding positions. Each location in location map indicates a 4×4 block, denoted by $Pal(f, i, j)$ or simply Pal, where f is the current frame number and $\{i, j\}$ indicates a 4×4 block in the i^{th} row and j^{th} column in a frame. The blocks that are not embedded are marked in the location map as $Pal = 0$. If coefficient in a block is modified to embed watermark bit then the location map is marked as $Pal = 2$, otherwise $Pal = 1$. The complete procedure is illustrated in the Algorithm 2 as follows.

Reversible Extraction

In the watermarked video, select first two coefficients denoted by A' and B' in a block, where $Pal \neq 0$. If A' is greater than B' then watermark bit is zero and the original unwatermarked value of A' is obtained by $|A| = |A'| - 2D'$, keeping signs of them unchanged when $Pal = 2$, or $A = A'$ when $Pal = 1$. Similarly, if B' is greater than A' then watermark bit is one and the original unwatermarked value of B' is obtained by $|B| = |B'| - 2D'$, keeping signs of them unchanged when $Pal = 2$ or $B = B'$ when $Pal = 1$. It is assumed that the location map is losslessly transmitted to the authorized client. The proposed reversible extraction algorithm is presented as Algorithm 3.

The proposed reversible watermarking method is elaborated using two simple illustrative examples.

Example 3 *Assume, the values of two nonzero AC coefficients (A and B) in a block are* 2 *and* 5*, respectively. The absolute difference between A and B will be*

$$D = |A - B| = 3$$

Algorithm 2: Reversible Embedding

Input: Original blocks

Output: Watermarked blocks and location map (Pal)

Select first two nonzero AC coefficients (A and B) in a block

$D = ||A| - |B||$

Initial value of Pal for every block is zero

if (watermark bit is 0 **and** $A < B$) **then**

$|A'| = |B| + D = |A| + 2D$

signs of A and A' is same

$Pal = 2$

elseif (watermark bit is 1 **and** $A > B$) **then**

$|B'| = |A| + D = |B| + 2D$

signs of B and B' is same

$Pal = 2$

else

$Pal = 1$

end

Algorithm 3: Reversible Extraction

Input: Watermarked blocks and location map (Pal)

Output: Unwatermarked blocks and extracted watermark

Select first two nonzero AC coefficients (A' and B') of a block in watermarked video

if $(Pal \neq 0)$ **then**

$D' = ||A'| - |B'||$

if $(|A'| > |B'|)$ **then**

watermark bit is 0

if $(Pal = 2)$ **then**

$|A| = |A'| - 2D'$

signs of A and A' is same

else

$A = A'$

end

else

watermark bit is 1

if $(Pal = 2)$ **then**

$|B| = |B'| - 2D'$

signs of B and B' is same

else

$B = B'$

end

end

end

If watermark bit is 0 *then the modified coefficients will be*

$$A' = A + 2 \times D = 8$$

and B *remains unchanged. The value of the location map for the block will be*

$$Pal = 2$$

During the extraction process at the decoder, if $A' > B'$ *and* $Pal = 2$ *then the watermark bit is* 0*. The absolute difference between* A' *and* B' *will be*

$$D' = |A' - B'| = 3$$

Reversibility of the watermarked coefficient to its original value is obtained by

$$A = A' - 2 \times D' = 2$$

and B *remains unchanged. The above example illustrates the watermark embedding and its reversible extraction when the watermark bit is zero.*

Example 4 *Assume, the values of two nonzero AC coefficients (*A *and* B*) in a block are* 2 *and* 5*, respectively. If watermark bit is* 1 *then the embedding rule [Eq. (4.8)] is satisfied, so no coefficients will be changed and the value of watermarked coefficients will be*

$$A = A' = 2 \text{ and } B = B' = 5$$

The value of the location map for the block will be

$$Pal = 1$$

During the extraction process at the decoder, if $A' < B'$ *and* $Pal = 1$ *then the watermark bit is* 1*. Reversibility of the watermarked coefficient to its original value is obtained by*

$$A' = A = 2 \text{ and } B' = B = 5$$

This example illustrates the watermark embedding and its reversible extraction when the watermark bit is one.

Robustness of the Proposed Reversible Method

In the proposed watermarking method, a robust reversible embedding method has been deployed. Robustness of the proposed reversible method are compared with an state of art reversible watermarking method [Tia03].

In difference expansion method [Tia03], first two coefficients in a block (denoted by x and y) are selected. Assume, average and difference between x and y are t and h, respectively. The embedding rule for difference expansion method [Tia03] is given as follows:

$$\begin{aligned} t &= \left\lfloor \frac{x+y}{2} \right\rfloor, \quad h = |x-y|, \\ h' &= 2 \times h + \text{Watermark Bit}, \\ x' &= t + \left\lfloor \frac{h'+1}{2} \right\rfloor, \quad y' = t - \left\lfloor \frac{h'}{2} \right\rfloor, \end{aligned} \tag{4.9}$$

where modified values of h, x, and y are denoted by h', x', and y', respectively. LSB of difference h' is replaced by the watermark bit [Eq. (4.9)]. However, any transformations (like blurring or geometric transformation) and lossy compressions (like H.264) can easily destroy LSBs of coefficients [SP96]. The extraction rule for difference expansion method [Tia03] is given as follows:

$$\begin{aligned} t' &= \left\lfloor \frac{x'+y'}{2} \right\rfloor, \quad h' = |x'-y'|, \\ \text{Watermark Bit} &= \text{Least Significant Bit}(h'), \\ h = \left\lfloor \frac{h'}{2} \right\rfloor \quad x &= t' + \left\lfloor \frac{h+1}{2} \right\rfloor, \quad y = t' - \left\lfloor \frac{h}{2} \right\rfloor, \end{aligned} \tag{4.10}$$

where modified t is denoted by t' and the watermark bit is extracted from LSB of h' [Eq. (4.10)]. Therefore, if LSB of h', *i.e.*, either x' or y' is changed, watermark bit will not be extracted properly.

In the proposed embedding rule [Eq. (4.8)], watermark bit is extracted based on

$|A| > |B|$ and $|B| < |A|$. The absolute difference between A and B either remains approximately same after embedding. This implies that the proposed watermarking method does not depend on LSB of A and B or LSBs of absolute difference between A and B. In other words, unlike least significant based reversible methods (*e.g.* difference expansion method [Tia03]), the proposed reversible method is robust against common signal processing attacks (like blurring) and lossy compressions (like H.264). Robustness of the proposed method increase with the increase in value of absolute difference between A and B. An illustrative example is given to show that the proposed method is robust against LSB modification attack even for very small of absolute difference between A and B (let it be 2).

Example 5 *Assume, the values of A and B are 3 and 1, respectively. If watermark bit is 0 then no change is required. Now, if watermark bit is 1 then $B' = 5$.*
Any LSB modification attacks can change the values of A and B to $\{(3,0),(2,1),(4,1),(3,2)\}$, when watermark bit is zero or $\{(3,4),(2,5),(4,5),3,6)\}$, when watermark bit is one. It is clear from results that changed values of A and B give the proper extraction of the watermark bit at the decoder obeying the proposed embedding rule [Eq. (4.8)].

Robustness of the proposed reversible method can be further increased by restricting the values of A and B by introducing a threshold. Furthermore, to keep the visual quality in acceptable limit suitable blocks for embedding can be selected using a threshold. It is observed from experimental results that there will be no significant increase in video bit rate if zero coefficients of the residual error block are not altered.

4.3 Proposed Method

In the proposed method, a bipolar watermark sequence (0,1) is embedded in the luminance component of 4×4 inter predicted blocks of P-frames. Candidate blocks are selected from the blocks selected using spatial and temporal masks ($MASK_T = 1$ and

$MASK_S = 1$) along with watermarking thresholds (V_T and R_T). In this section, first the watermark embedding and extraction methods are described. Then, the reversible watermark extraction with coefficient recovery algorithm is presented.

4.3.1 Watermark Embedding

In the proposed method, watermark embedding is performed in nonzero quantized AC coefficients in 4×4 blocks of P-frames. The DC coefficients are not used for embedding as perturbing the value of DC coefficients may cause higher visual artifacts. Zero valued AC coefficients are also not used because there is a chance of increase in the video bit rate if zero valued coefficients become nonzero due to embedding as mentioned in Section 3.2.2.

The step by step description of the proposed watermark embedding method is as follows:

Step 1: P-frames are divided into 4×4 non-overlapping blocks.

Step 2: The blocks suitable for watermark embedding are selected using spatial and temporal analysis, where $MASK_T$ and $MASK_S$ are unity and watermarking thresholds as described in Section 4.2.1, Section 4.2.2, and Section 4.2.3, respectively.

Step 3: Candidate blocks are selected for embedding from all blocks selected in Step 2 using a pseudo random key. The location of the candidate blocks is saved in a location map. Each location in location map indicates a 4×4 block, denoted by $Pal(f, i, j)$ or simply Pal, where f is the current frame number and $\{i, j\}$ indicates a 4×4 block in the i^{th} row and j^{th} column in a frame. The initial value of the Pal is zero.

Step 4: Estimation of different values in odd and even sequences is performed.

Step 4.1: The sum of absolute values of coefficients in odd and even sequences, *i.e.*, Odd_{Sum} and $Even_{Sum}$ for a candidate block is calculated using Eq. (4.3) and Eq. (4.4), respectively.

Step 4.2: Nonzero highest coefficient of odd and even sequences is estimated using Eq.

(3.6) and Eq. (3.6), respectively.

Step 4.3: The absolute difference ($diff_{eo}$) between Odd_{Sum} and $Even_{Sum}$ is calculated using Eq. (4.5).

Step 5: Each candidate block is embedded using a variant of Algorithm 2.

The complete embedding algorithm is described in Algorithm 4.

If watermark bit is zero (0) and $Odd_{sum} < Even_{sum}$ then the absolute value of the modified highest coefficient in odd sequence ($|AC'_o|$) is $|AC_o| + 2 \times diff_{eo}$ and signs of AC'_o and AC_o is same. Similarly, if watermark bit is one (1) and $Odd_{sum} > Even_{sum}$ then the absolute value of the modified highest coefficient in even sequence ($|AC'_e|$) is $|AC_e| + 2 \times diff_{eo}$ and signs of AC'_e and AC_e is same. The value of the location map for that block is $Pal = 2$. If watermark bit is zero (0) and $Odd_{sum} > Even_{sum}$ or watermark bit is one (1) and $Odd_{sum} < Even_{sum}$, then no modification is performed and location map is marked as $Pal = 1$.

The location map generated in Step 2 is sent to the decoder. The block diagram in Figure 4.2 describes the overall embedding process of the proposed watermarking method.

The proposed method can be implemented for all P-frames. Similar to the Section 3.2.6, a subset of P-frames can be selected for embedding to increase security of the proposed method. This subset of P-frames can be selected randomly. Another pseudo random number generator can be used for this purpose.

4.3.2 Watermark Extraction

Watermark extraction is performed at the decoder after entropy decoding. The location of candidate blocks where the watermark is embedded is saved in a location map during the embedding process. This location map is used during the extraction process at the decoder to select candidate blocks. The embedded watermark bit is extracted from each candidate block as follows:

Algorithm 4: Watermark Embedding

Input: Candidate blocks in P-frame

Output: Watermarked blocks and location map (Pal)

for each candidate block in P-frames **loop**

Odd_{Sum} and $Even_{Sum}$ are estimated using Eq. (4.3) and Eq. (4.4), respectively

$diff_{eo} = |Odd_{sum} - Even_{Sum}|$

AC_o and AC_o are obtained using Eq. (3.6) and Eq. (3.6), respectively

if (watermark bit is 0 **and** $Odd_{sum} < Even_{sum}$) **then**

$|AC'_o| = |AC_o| + 2 \times diff_{eo}$

signs of AC'_o and AC_o is same

$Pal = 2$

elseif (watermark bit is 1 **and** $Odd_{sum} > Even_{sum}$) **then**

$|AC'_e| = |AC_e| + 2 \times diff_{eo}$

signs of AC'_e and AC_e is same

$Pal = 2$

else

$Pal = 1$

end

end

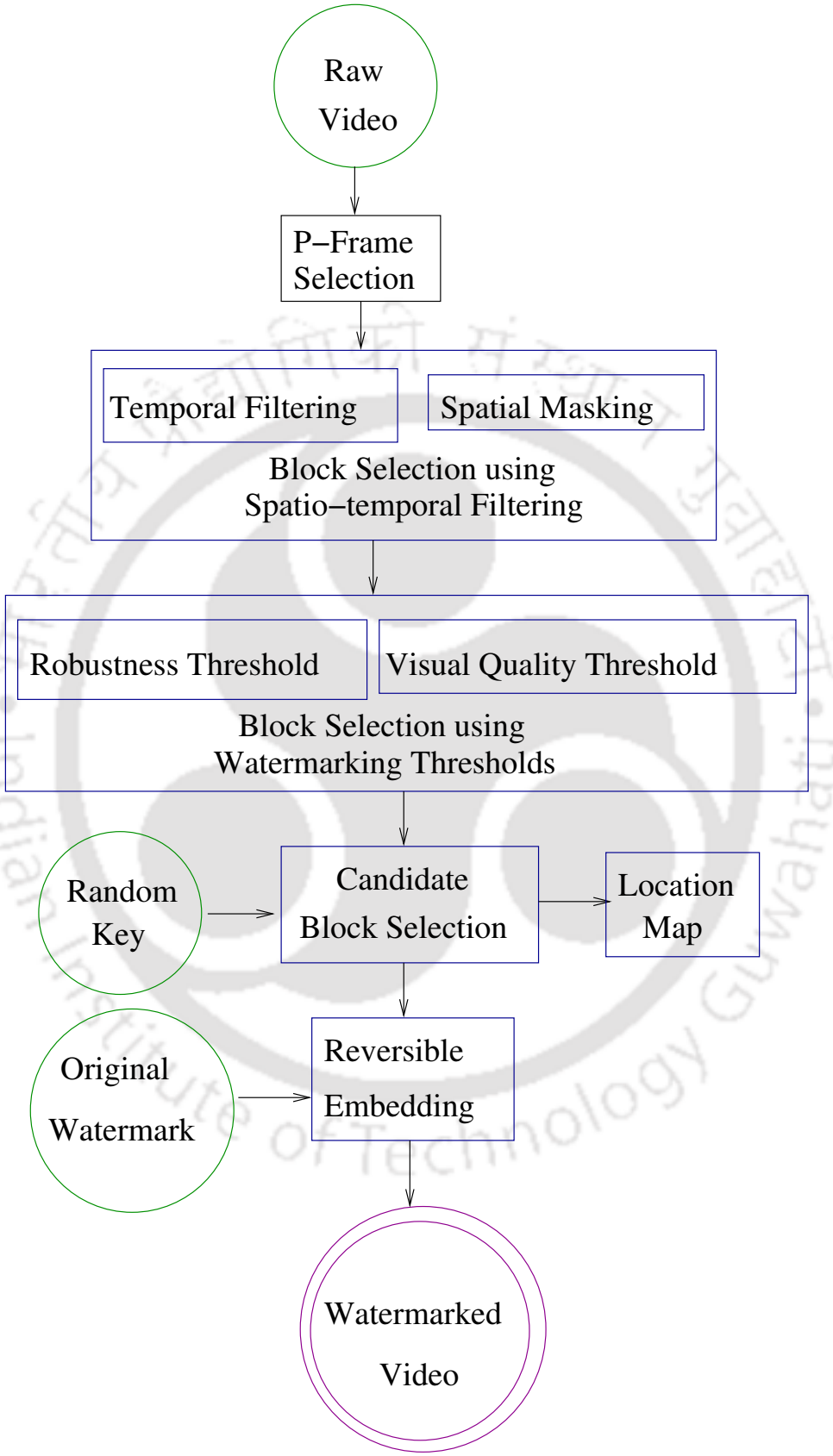


Figure 4.2 The block diagram of the proposed embedding method.

$$\text{watermark bit} = \begin{cases} 0 & Odd'_{sum} > Even'_{sum} \\ 1 & \text{otherwise,} \end{cases} \tag{4.11}$$

where Odd'_{sum} and $Even'_{sum}$ are sum of the absolute values of the coefficients in odd and even sequences of a block in a P-frame of the watermarked video. The block diagram in Figure 4.3 shows the overall extraction process of the proposed watermarking method.

4.3.3 Watermark Extraction with Coefficient Recovery

Recovery of the original coefficients from the watermarked coefficients is performed using a variant of the reversible extraction method mentioned in Algorithm 3. Candidate blocks are found using location map. AC'_o, AC'_e, Odd'_{sum}, and $Even'_{sum}$ are nonzero highest coefficients and sum of the absolute values of the coefficients in odd and even sequences, respectively of a candidate block in a P-frame of the watermarked video.The complete reversible extraction algorithm is given in Algorithm 5:

If $Odd'_{sum} > Even'_{sum}$ then watermark bit is zero and if $Pal = 2$ then original unwatermarked value of the highest coefficient in the odd sequence (AC_o) is $AC'_o - 2 \times diff'_{eo}$ and signs of AC'_o and AC_o is same. If $Pal = 1$, then $AC_o = AC'_o$ and $AC_e = AC'_e$. Similarly, if $Odd'_{sum} < Even'_{sum}$ then watermark bit is one and if $Pal = 2$ the original unwatermarked value of the highest coefficient in the even sequence (AC_e) is $AC'_e - 2 \times diff'_{eo}$ and signs of AC'_e and AC_e is same. Modified coefficient (AC'_o or AC'_e) is reset to its original value (AC_o or AC_e). Original blocks are recovered through reversible process, so future blocks are predicted from original unwatermarked blocks rather than embedded ones. This helps to prevent distortion drift.

4.4 Salient Features of the Proposed Method

In this section, characteristics of the proposed method, such as robustness, visual quality, security, embedding capacity, and complexity overhead of the proposed method, are

Algorithm 5: Watermark Extraction with Coefficient Recovery

Input: Candidate blocks in watermarked video and location map (Pal)

Output: Unwatermarked blocks and extracted watermark

for each candidate block ($Pal \neq 0$) in watermarked video **loop**

$diff'_{eo} = ||Odd'_{sum}| - |Even'_{sum}||$

if ($|Odd'_{sum}| > |Even'_{sum}|$) **then**

watermark bit is 0

if ($Pal = 2$)

$|AC_o| = |AC'_o| - 2 \times diff'_{eo}$

signs of AC'_o and AC_o is same

else

watermark bit is 1

if ($Pal = 2$)

$|AC_e| = |AC'_e| - 2 \times diff'_{eo}$

signs of AC'_e and AC_e is same

end

end

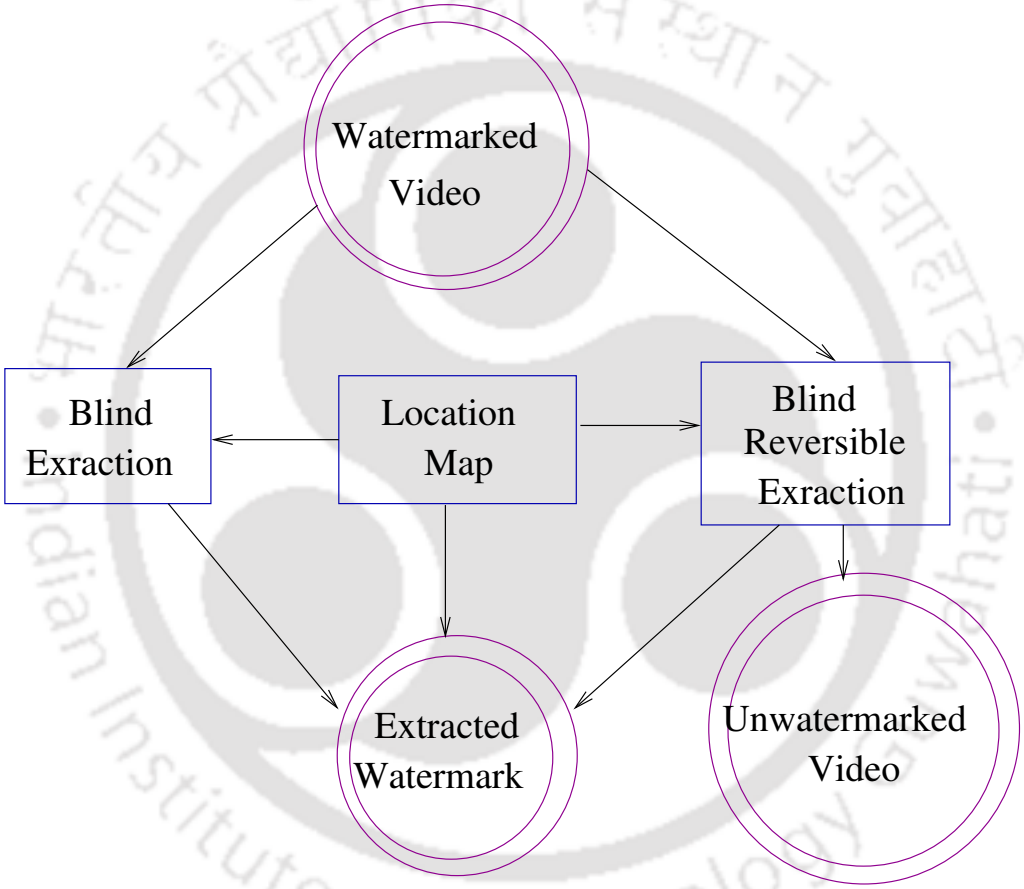


Figure 4.3 The block diagram of the proposed extraction method.

discussed.

4.4.1 Robustness of the Proposed Method

In Section 4.2.4, robustness of the proposed reversible method is shown. Robustness of the proposed method are enhanced by incorporating the robustness threshold (R_T) [MAAK10], where R_T is a positive integer. Block selection criteria are further modified in Eq. (4.6) to incorporate the robustness threshold such that

$$R_T < diff_{eo}$$

Eq. (4.6) implies that the absolute difference between $|AC_o|$ and $|AC_e|$ is greater than R_T. If watermark bit is one and $Odd_{sum} > Even_{sum}$ then according to Algorithm 4:

$$|AC'_e| = |AC_e| + 2 \times diff_{eo}$$

In watermarked video, the absolute difference between the watermarked coefficients of a block in watermarked video is $diff_{eo}$. After watermark embedding, the absolute difference between $|AC'_o|$ and $|AC'_e|$ is $diff_{eo}$. If $diff_{eo}$ is very small (let $diff_{eo} = 1$) then changing LSB of $|AC'_o|$ or $|AC'_e|$, will make the detection of watermark bit at the decoder impossible. Therefore, R_T is incorporated to increase robustness.

Robustness of the proposed method increase with higher value of $diff_{eo}$. Again, R_T will be large for higher values of $||AC'_o| - |AC'_e||$. In case of attacks, for destroying a watermark bit the attacker need to guess the value of AC_o, AC_e, and/or $diff_{eo}$, which is specific to a block and random across the blocks. This implies that increasing the value of R_T will increase the robustness of the proposed reversible method. The same is true when watermark bit is zero. Selection of the appropriate value of R_T is important. The relation of R_T with the robustness and the number of blocks available for embedding

is given as follows:

$$R_T \propto \frac{\text{robustness}}{\text{number of blocks available for embedding}}$$

The value of R_T is determined based on exhaustive results. The change in embedding capacity and robustness (against recompression error) [refer to Appendix] of the watermarked video are measured (refer to Algorithm 4) with the change in R_T is depicted in Table 4.1. If R_T increases, robustness increase, but the embedding capacity decreases. If R_T decreases, robustness decrease, but embedding capacity increases. Therefore, a trade-off exists between embedding capacity and robustness to select the value of R_T. The value R_T is considered as a small positive integer.

Table 4.1 Selection of robustness threshold R_T based on PSNR and Embedding Capacity, when $V_T = 11$.

Video Sequence	R_T	Average Robustness	Average embedding capacity/frame
Carphone	1	86	31
	3	91	27
	5	94	23
Foreman	1	82	49
	3	89	41
	5	93	35
News	1	82	39
	3	90	35
	5	93	33
Salesman	1	87	36
	3	95	29
	5	96	20
Suzie	**1**	88	47
	3	90	40
	5	91	33
Trevor	1	89	29
	3	93	26
	5	95	21

4.4.2 Visual Quality of the Proposed Method

The embedding of invisible watermark is performed on P-frames for better visual quality. In the proposed embedding method, to increase the sum of odd or even sequence in a block, the value of the highest coefficient of that sequence is increased (refer to Algorithm 4). If $|Odd_{sum} - Even_{sum}|$ is very large then degradation in visual quality may be more. Therefore in Eq. (4.6), visual quality threshold (V_T) is incorporated to minimize significant visual quality degradation such that

$$diff_{eo} < V_T$$

where V_T is a positive integer. If V_T is small then addition of extra noise due to embedding will be less and thus visual quality degradation will be less. However, the number of blocks available for embedding will decrease with a smaller value of V_T. The relation of V_T with the perceptual quality of the watermarked video and the number of blocks available for embedding is given as follows:

$$V_T \propto \frac{\text{number of blocks available for embedding}}{\text{perceptual quality}}$$

In the proposed method, V_T is determined based by an exhaustive set of experiments, given in table 4.2.

4.4.3 Embedding Capacity of the Proposed Method

Embedding capacity of the proposed method depends on block selection criteria like spatial and temporal masking ($MASK_S$ and $MASK_T$), visual quality threshold (V_T), and robustness threshold (R_T). Total number of blocks in a P-frame, average number of blocks filtered out after spatial and temporal masking, number of blocks selected after applying the watermarking thresholds, and average number of blocks available for embedding in different video sequences are depicted in Figure 4.4. The relation of em-

Table 4.2 Selection of visual quality threshold V_T based on PSNR and Embedding Capacity, when $R_T = 3$.

Video Sequence	V_T	Average PSNR	Average Embedding Capacity
Carphone	7	37.7	21
	11	37.39	28
	15	37.09	31
Foreman	7	36.78	26
	11	36.39	41
	15	35.89	45
News	**7**	37.72	35
	11	37.19	37
	15	36.91	40
Salesman	7	36.50	19
	11	36.28	30
	15	35.95	33
Suzie	7	37.62	37
	11	37.19	42
	15	36.08	53
Trevor	7	37.01	21
	11	36.95	26
	15	36.89	31

bedding capacity with embedding parameters are elaborated in succeeding paragraphs.

Embedding capacity is proportional to temporal threshold MV_{th} (refer to Section 4.2.1).

$$\text{Embedding Capacity} \propto MV_{th}$$

For spatial analysis, embedding capacity is inversely proportional to NNZ threshold NNZ_{th} (refer to Section 4.2.2).

$$\text{Embedding Capacity} \propto \frac{1}{NNZ_{th}}$$

Intuitively, embedding capacity is inversely proportional to robustness threshold R_T (refer to Section 4.3.1) and proportional to visual quality threshold V_T (refer to Section 4.3.1). Hence, the embedding capacity is related to aforementioned embedding

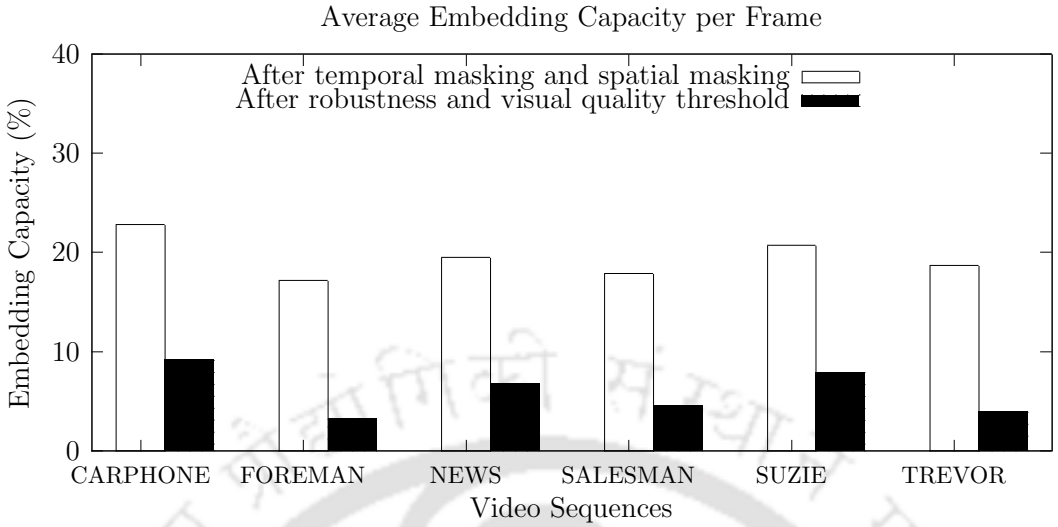


Figure 4.4 Average number of blocks suitable for embedding.

parameters as follows:

$$\text{Embedding Capacity} \propto \frac{MV_{th} \times V_T}{NNZ_{th} \times R_T}$$

4.4.4 Security

In the proposed method, selected blocks for embedding are based on spatial and temporal masks ($MASK_S$ and $MASK_T$) and watermarking parameters (V_T and R_T). Similar as Section 3.2.6, candidate blocks are from these selected blocks (selected randomly using a pseudo random key). Furthermore, P-frames are selected randomly using another pseudo random number generator. Dual private pseudo random key is used the to select candidate blocks.

4.4.5 Complexity and Overhead

In the proposed method, the partially decoded video coding parameters are used for watermarking which decreases the complexity of the proposed method than any uncompressed domain methods. However, embedding locations are sent as side information to the decoder which reduces the computational complexity regarding embedding region

selection at the time of extraction in decoder. The size of the location map after run-length encoding is tabulated in in Table 4.3 where it is observed that the size of side information is negligible in comparison with the size of compressed video. However, the

Table 4.3 Average Size of location map Per Watermarked Video.

Sequence	Compressed Video Size (KB)	location map Size (KB)
Carphone	51	0.199
Foreman	67	0.204
News	54	0.201
Salesman	52	0.100
Suzie	55	0.201
Trevor	64	0.203

size of the location map is increased compared to the previous algorithm in Chapter 3, this increase in size is negligible compared to the size the compressed video. In the proposed reversible watermarking method, reversibility is achieved at the cost of little increase in the size of location map a during the embedding process.

4.5 Experimental Results

The proposed method is implemented using H.264/AVC [Ric10] reference software JM 17.2 [S08]. Table 4.4 shows the experimental setup. The minimum value of R_T would be 3 based on the experimental result given in Table 4.1. The minimum number of NNZ required is 2 to embed a watermark bit using the Algorithm 4. From this observation, the value of NNZ_{th} is set as 2. It is observed from Table 4.2 that if the absolute difference between the selected coefficients in a block is less than equal to 10 then no significant visual artifacts is seen in watermarked videos. Therefore V_T is taken as 11. Intuitively, watermark embedding in low motion region creates less temporal flicker. It is experimentally observed that if the temporal threshold (MV_{th}) is within 20 then no temporal flicker can be noticed in watermarked videos as shown in Table 4.5. The simulation results for the visual quality of the watermarked video, increase in video bit rate, and robustness against different attacks are shown in Section 4.5.1 and

Table 4.4 Experimental Setup

Parameters	**Values**
Video Format	Quarter Common Intermediate format (QCIF)
Frame Resolution	176×144
Frame Rate	30 frames per second
Codec Used	H.264/AVC reference software JM 17.2
Intra Period	10
GOP Structure	IBBPBBPBBP
Encoding Profile	High Profile
Entropy Encoding	CAVLC
Number of frames encoded	100 frames per video
Quantization Parameter (QP)	28
Payload	{100, 150, 200, 250, 300}
Video Sequence	Carphone, Foreman, News, Salesman, Suzie, Trevor
NNZ Threshold (NNZ_{th})	2
Temporal Threshold (M_{th})	20
Visual Quality Threshold (V_T)	11
Robustness Threshold (R_T)	3

Section 4.5.2, respectively.

Table 4.5 Selection of temporal threshold MV_{th} based on PSNR by embedding one bit per 4×4 selected block.

MV_{th}	Average PSNR					
	Carphone	Foreman	News	Salesman	Suzie	Trevor
0	36.72	35.32	36.79	35.52	35.76	35.64
5	37.61	36.82	37.78	37.02	37.68	36.95
10	37.54	36.67	37.62	36.89	37.21	36.89
15	37.41	36.44	37.39	36.53	36.67	**36.8**
20	37.25	**36.18**	**37.2**	**36.19**	36.31	36.47
25	**36.99**	35.79	36.93	35.87	**36.05**	36.02

4.5.1 Visual Quality and Bit Increase Rate

The visual quality of watermarked video is compared with recent existing literature with respect to Video Quality Metric (VQM) and Peak Signal-to-Noise Ratio (PSNR). Bit Increase Rate (BIR) is also calculated and compared with recent existing methods.

VQM, PSNR, and BIR of the proposed method at payload={100, 150, 200, 250, 300} for an average of 95 frames for video sequences, such as, {Carphone, Foreman, News, Salesman, Suzie, and Trevor} are depicted in Table 4.6. VQM in Table 4.6 is in the order of 10^{-2}. It indicates that watermarked videos have acceptable visual quality. Moreover, PSNR also confirms that the proposed embedding method does not sacrifice the visual quality very much. It is also observed that the increase in video rate (BIR) is in the order of 10^{-3}, which can be regarded as nominal.

Table 4.6 Results for VQM, PSNR, and BIR of the proposed method.

Sequence	Payload	VQM$\times 10^{-2}$	PSNR (dB)	BIR$\times 10^{-3}$
Carphone	100	2.32	37.77	1.6
	150	2.72	37.56	1.66
	200	2.96	37.34	0.65
	250	3.11	37.3	1.2
	300	4.39	37.15	0.74
Foreman	100	2.91	36.9	4.4
	150	2.95	36.78	2.2
	200	4.12	36.43	0.49
	250	4.86	36.04	0.57
	300	3.36	35.92	0.68
News	100	2.60	37.82	0.68
	150	2.69	37.6	0.97
	200	2.75	37.51	-0.08
	250	3.02	37.23	-4
	300	3.4	37.1	-0.47
Salesman	100	4.51	36.66	2.08
	150	4.60	36.44	1.9
	200	4.66	36.33	1.1
	250	4.71	36.04	1.6
	300	4.64	36.02	1.3
Suzie	100	3.65	37.78	2.7
	150	3.63	37.6	1.3
	200	3.61	37.41	2.0
	250	3.41	36.33	2.1
	300	3.20	36.11	1.2
Trevor	100	3.93	37.08	0.85
	150	3.59	37.01	1.3
	200	3.01	36.94	1.0
	250	3.10	36.86	1.9
	300	2.80	36.83	0.73

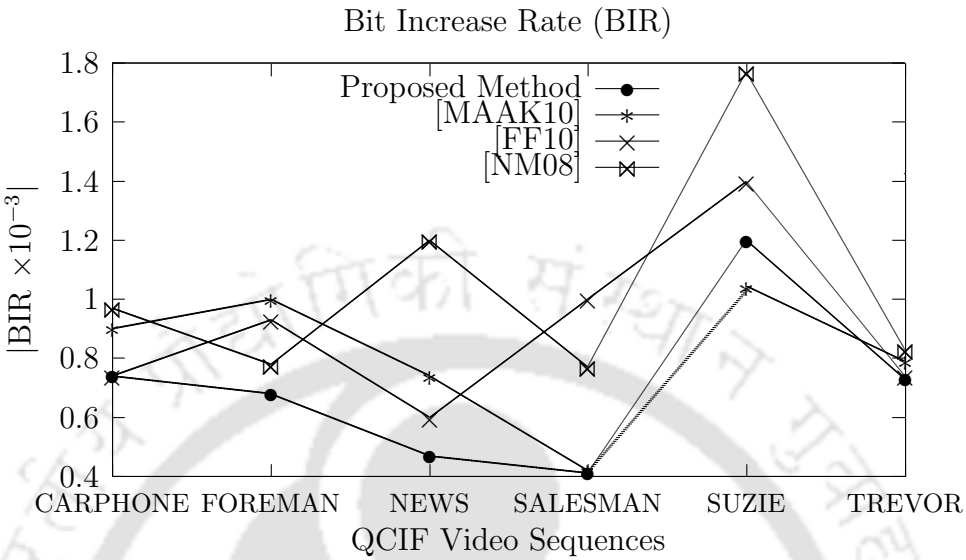


Figure 4.5 Average Bit Increase Rate (BIR).

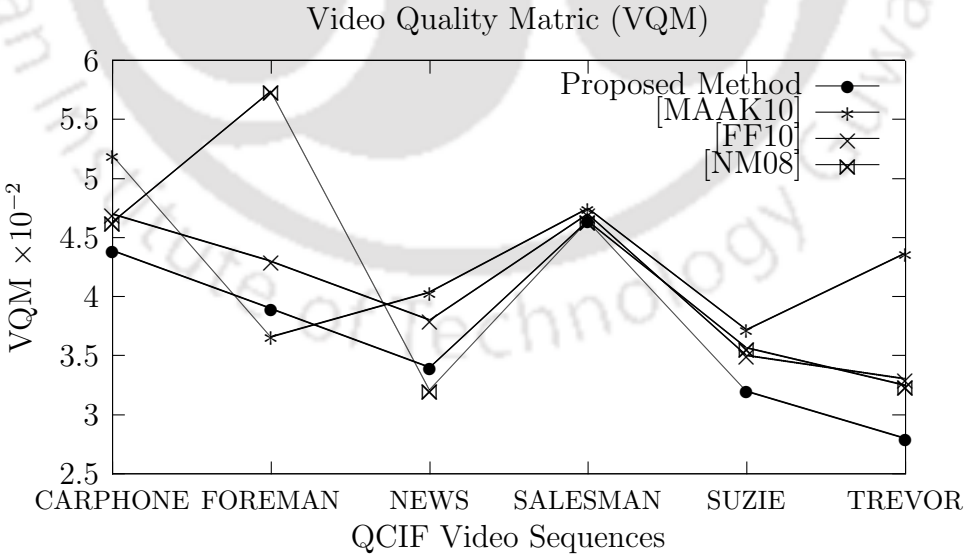


Figure 4.6 Average Video Quality Metric (VQM).

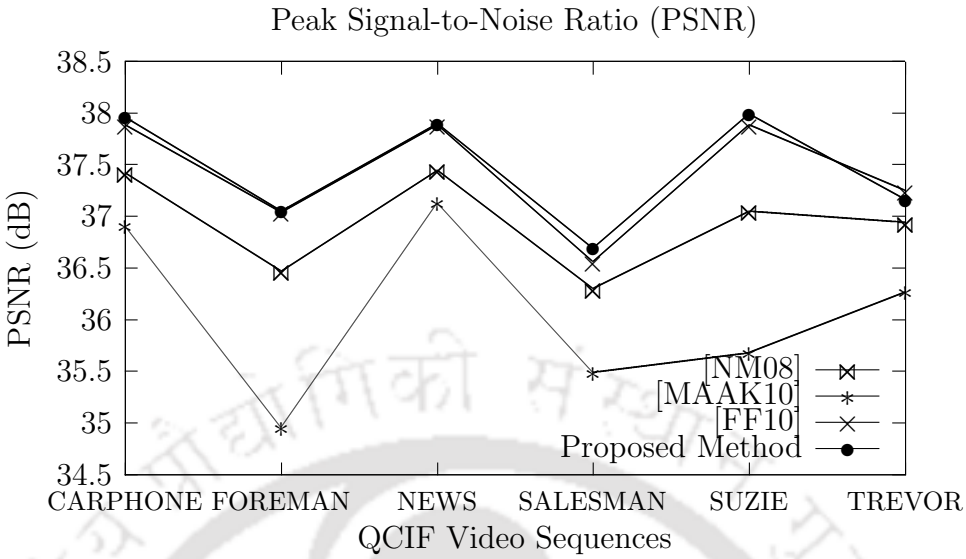


Figure 4.7 Average Peak Signal-to-Noise Ratio (PSNR).

The comparison of the proposed P frame based method with existing two P-frame based methods ([NM08, FF10]) and one I-frame based method [MAAK10] is given in Figure 4.5 for average Bit Increase Rate. Similar results for VQM and PSNR are depicted in Figure 4.6 and 4.7 respectively.

Figure 4.5 shows a nominal increase in video bit rate for the proposed method. Figure 4.6 and Figure 4.7 illustrate the superiority of the proposed method over the existing schemes [MAAK10, NM08, FF10] with respect to the VQM and PSNR, respectively.

The drift error propagation is more critical issue for videos having long GOPs or longer videos. Since we are using reversible watermarking technique, drift error is compensated even in videos with longer GOPs. Figure 4.8 shows that with the increase in size of GOP, the PSNR of the watermarked video decreases, denoted by average PSNR, but PSNR of the un-watermarked (after removal of reversible watermark), denoted by decoded PSNR, video remain comparable with the original PSNR of the Foreman video.

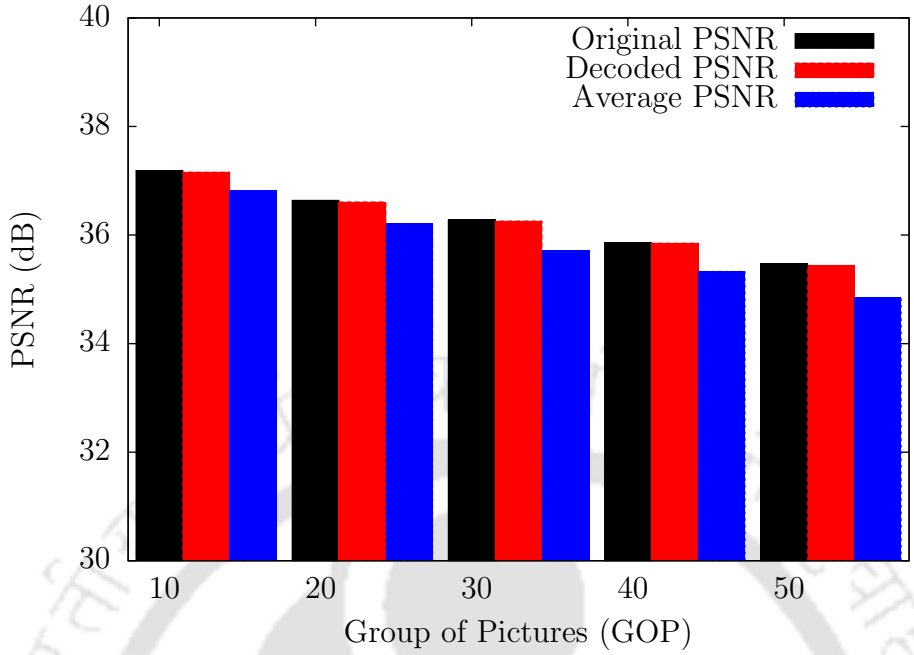


Figure 4.8 PSNR vs GOP for Foreman Video.

4.5.2 Robustness against attacks

The bit error rate (BER) is used for evaluating the robustness [refer to Appendix] of the proposed method against different attacks. The number of error bits is determined over all frames of watermarked video stream. As mentioned in Section 3.3.3, the information about the embedding locations which is saved as a location map during the embedding process is communicated to the decoder during the extraction process. In the absence of location map, if a watermarked location is not detected or an unwatermarked position is selected incorrectly, the synchronization in the watermark sequence may be lost. This decreases the robustness of the watermarking method. Figure 4.9 shows the robustness against changing QP from 28 to range of 20 to 36 and in presence of gaussian noise for the *foreman* video.

Robustness of the proposed method are compared with the methods in existing literature [MAAK10, FF10]. The results for changing the quantization parameter (QP) from 28 to 30 and from 28 to 26 are depicted in Figure 4.10 and Figure 4.11, respectively.

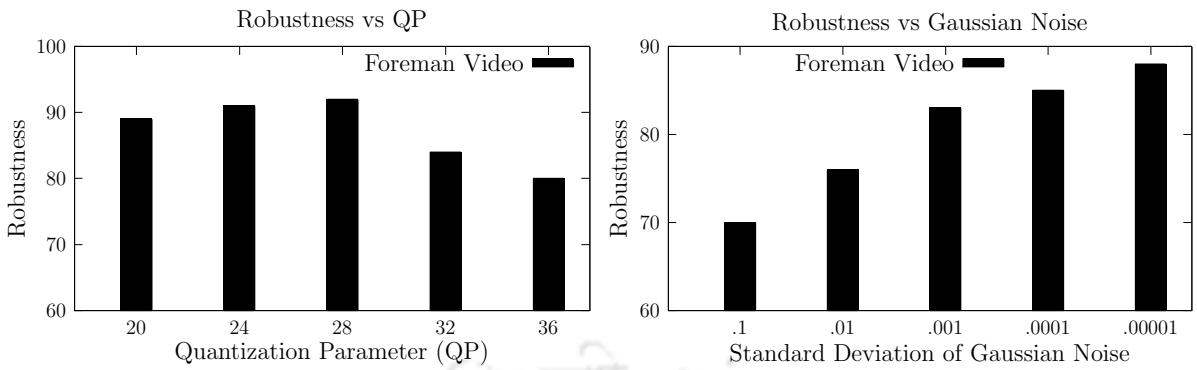


Figure 4.9 The change in robustness with QP and gaussian noise in the foreman video.

Similar result for recompression attack is presented in Figure 4.12. In [MAAK10], the number of nonzero coefficients (NNZ) in a block is decreased to embed a watermark bit. In [FF10], the watermark is extracted based on LSB of the embedding coefficients. In the proposed method, the absolute value of one nonzero coefficient is increased in a block and the watermark is extracted from the absolute difference between two highest value AC coefficients of two subsequences. Intuitively, this may be the reason for the proposed method to perform better for re-encoding (QP 28 to 30 and 26) and re-compression error.

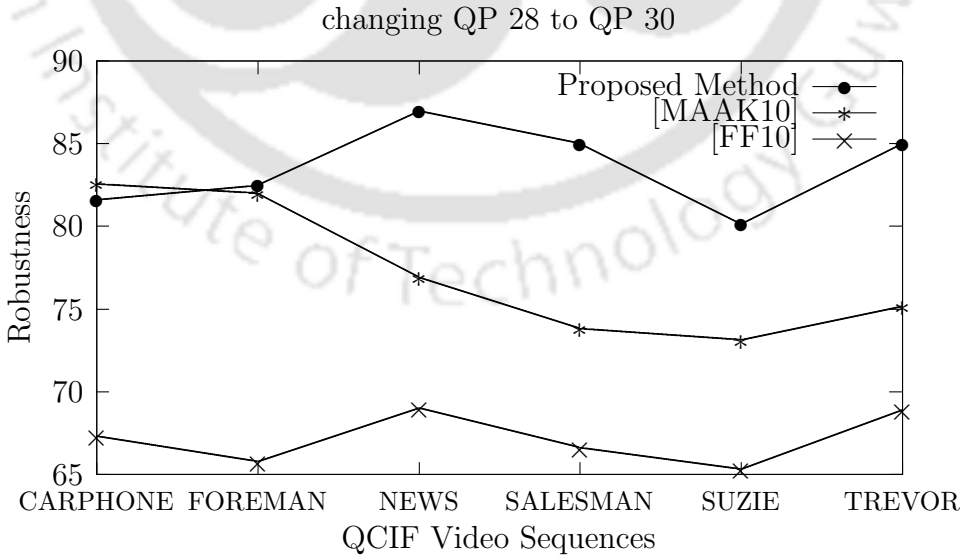


Figure 4.10 Average Bit Error Rate (BER) for changing QP from 28 to 30.

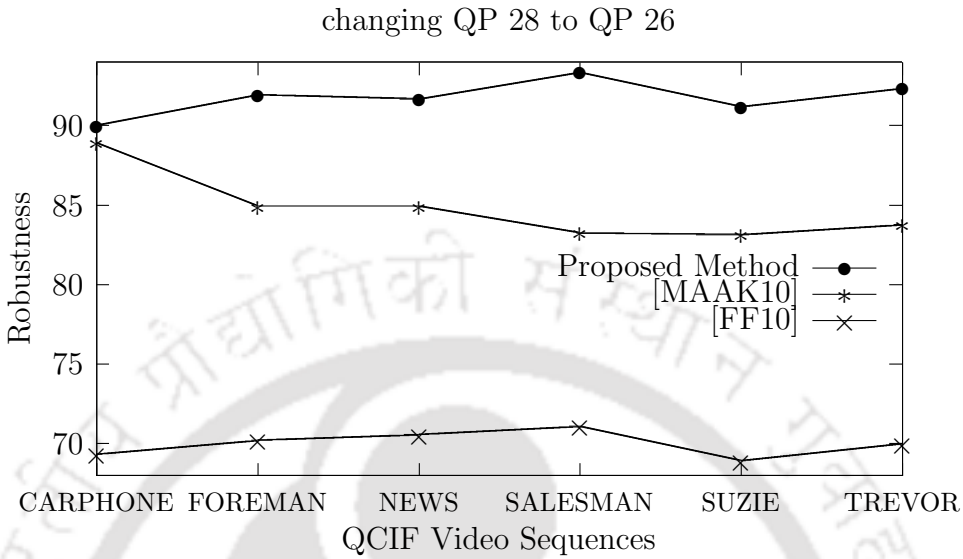


Figure 4.11 Average Bit Error Rate (BER) for changing QP from 28 to 26.

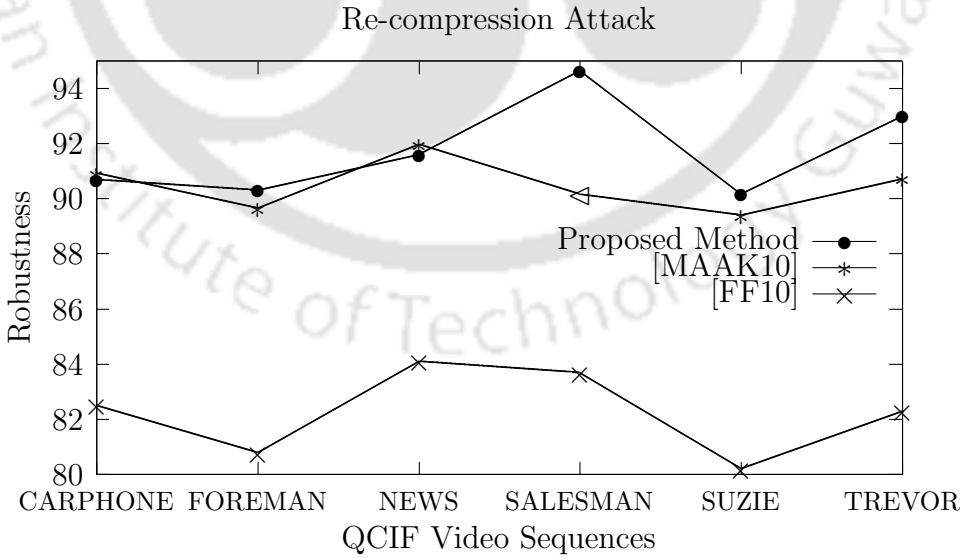


Figure 4.12 Average Bit Error Rate for Re-compression error.

Figure 4.13, Figure 4.14, Figure 4.15, and Figure 4.16 show the robustness against salt and pepper noise, gaussian filter, circular averaging filter, and additive white gaussian noise in comparison with the existing literature [MAAK10] and [FF10], where Gaussian Noise density = 0.001, Salt and Pepper Noise density = 0.001, Circular averaging filter r = 0.06, Gaussian Filter [5×5], and sigma=0.3 [MAAK10]. It is observed from the results that the proposed method outperforms the existing methods.

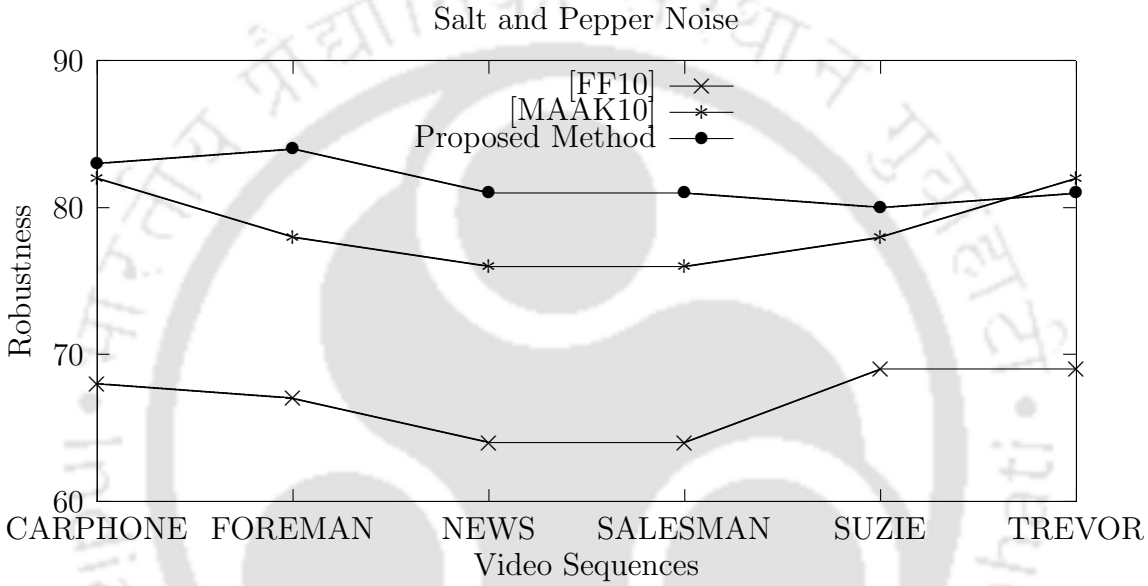


Figure 4.13 Average Bit Error Rate (BER) for salt and pepper noise.

In this work, Reed-Solomon code [SJM+04] [refer to Appendix] is incorporated with the proposed method to enhance its robustness against frame dropping, frame averaging, and frame swapping attacks. It is observed that the accuracy of watermark detection against common image and video processing attacks are also enhanced.

In the experimentation, 95 frames per video is considered. As per the structure of the GOP in Table 6.1, 95 frames will have 30 P-frames considering all P-frames are used for embedding. For 20% frame recovery (*i.e.*, on an average of 6 P-frames out of 30 P-frames) against frame dropping, averaging and swapping (FDAS) attacks, some extra bits are added with watermarked video sequence. This increases the video bit rate. Figure 4.17 shows that the increase in the video bit rate is still nominal and gives

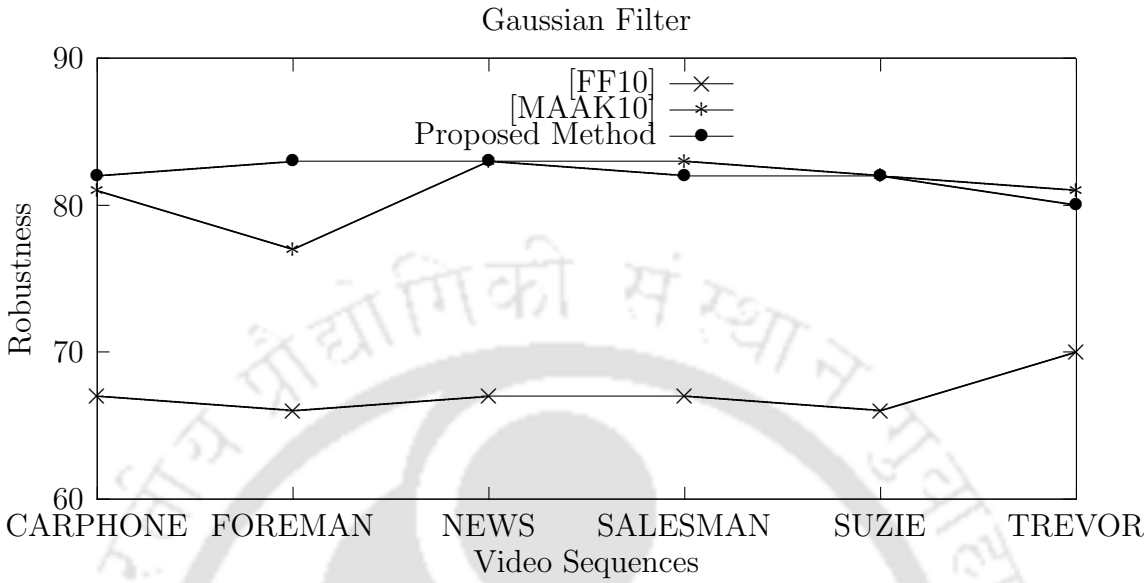


Figure 4.14 Average Bit Error Rate for gaussian filter.

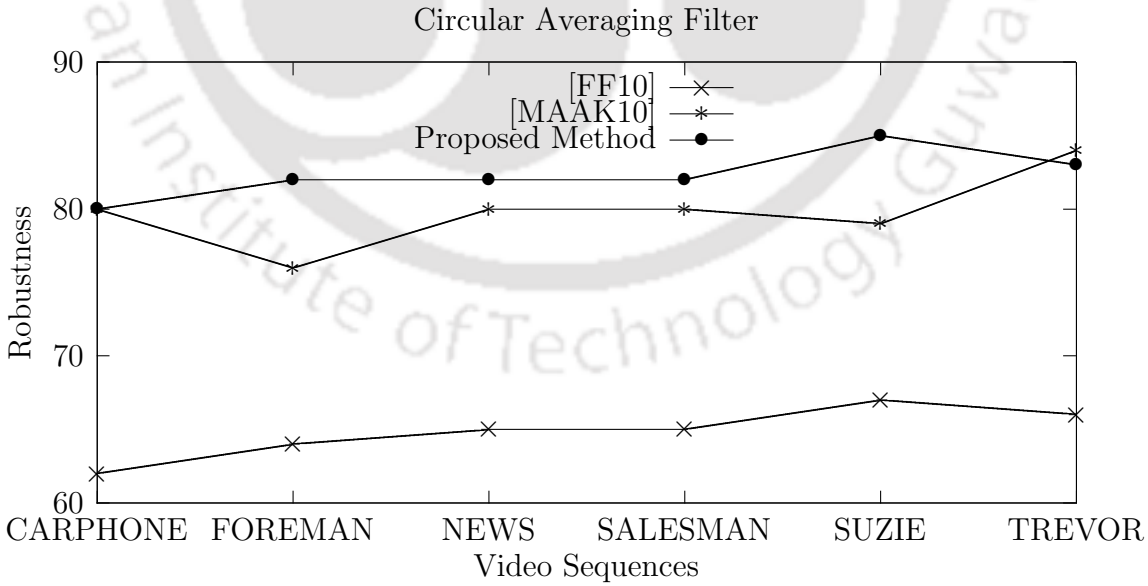


Figure 4.15 Average Bit Error Rate (BER) for circular averaging filter.

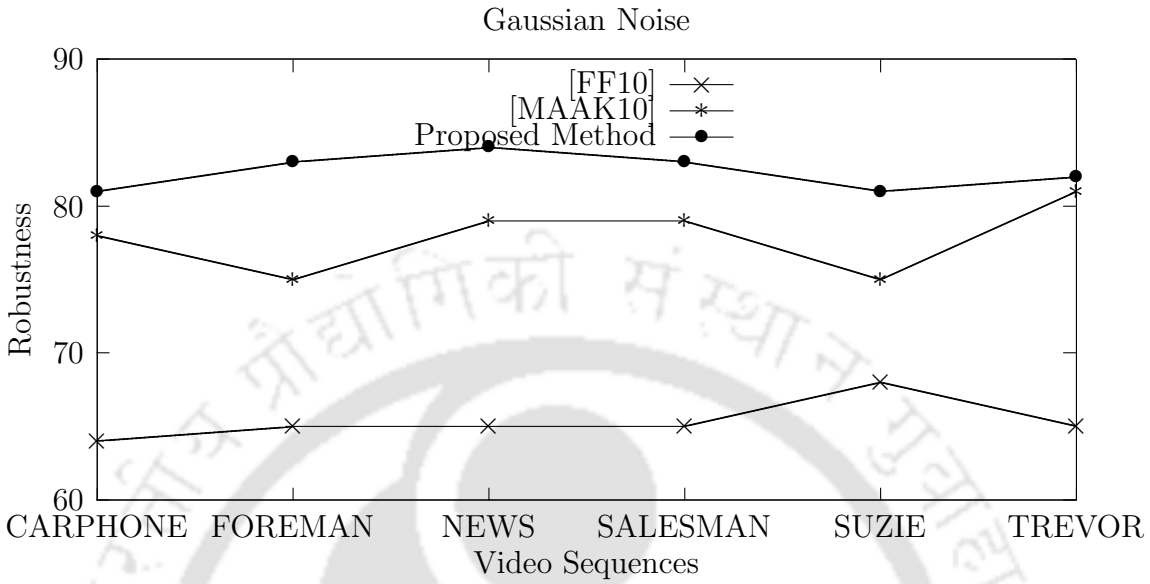


Figure 4.16 Average Bit Error Rate for gaussian noise.

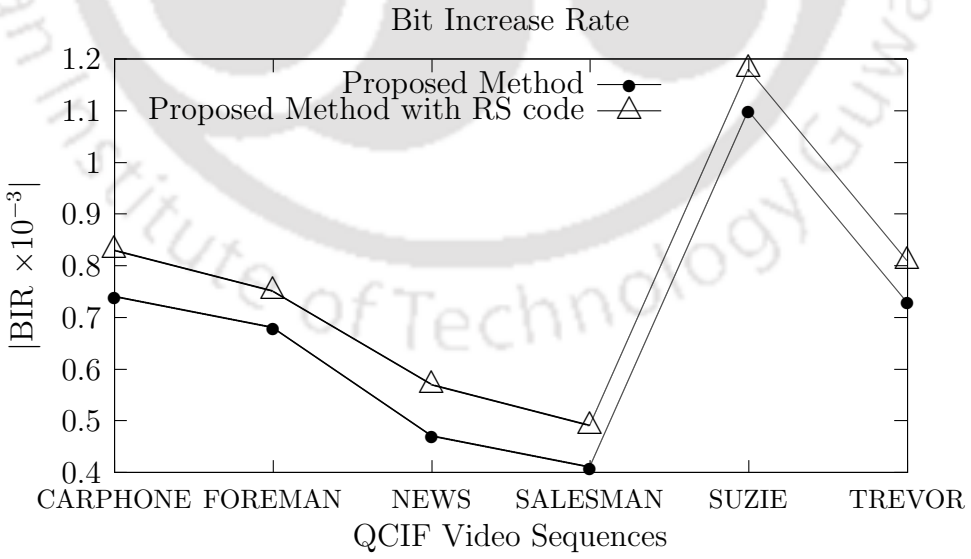


Figure 4.17 Average Bit Increase Rate (BIR)

acceptable results for most of the videos. The use of Reed-Solomon code also enhances the accuracy of watermark detection against common image processing attacks and re-encoding attacks by 20%. The complete (100%) detection of watermark sequence is possible at the decoder for most of the video sequences.

4.6 Summary

In this chapter, a robust video watermarking method with blind extraction for H.264/AVC compressed video streams has been proposed, which can present drift distortion. The chapter ensures acceptable visual quality and marginal increase in video bit rate by searching suitable blocks for embedding using spatial as well as temporal analysis. Drift compensation is avoided using the reversible watermarking technique. Moreover, robustness of the proposed method are enhanced using the proposed robust reversible watermarking method. Robustness threshold and visual quality threshold are used to control the trade-off among robustness, visual quality and payload. It is experimentally shown that proposed method outperforms other existing methods with respect to visual quality, bit increase rate and robustness against different attacks.

Watermark embedding in I-frames are not prone frame dropping, averaging, and swapping attacks. Moreover, suitable blocks for embedding in I-frames are more than P-frames. However, the main disadvantage of I-frame watermarking is high degradation in perceptual quality. In the next chapter, an efficient algorithm is designed for I-frame embedding maintaining higher perceptual quality and lower increase in video bit rate. The location aware and unaware detection of watermark bits at the decoder are also estimated.

Chapter 5

Robust Watermarking in I-frame

In the last two chapters, P-frames are used for watermark embedding in compressed domain. The main reason behind the selection of P-frames is to have better perceptual quality as shown in Chapter 3. However, watermark embedding in P-frames are prone to frame dropping, averaging, and swapping (FDAS) attacks. The watermarking in I-frames does not face such attacks in general because FDAS attack on I-frames will subsequently degrade the perceptual quality of watermarked video with respect to the Human Visual System (HVS). In addition, I-frame have more nonzero coefficients than the P-frames or the B-frames.

It is observed in Chapter 1 that many attempts have been made on I-frame based watermarking in the H.264/AVC encoded video. Mansouri *et. al.* in [MAAK10] have illustrated that luminance 4×4 intra predicted mode blocks generally represents the busy regions whereas 16×16 mode blocks represent smooth regions. Intuitively, embedding in busy regions of the video frame may cause relatively less visual artifacts. Authors have further shown that the degree of synchronization noise reduces when embedding is done in a 4×4 block having higher number of nonzero coefficients (NNZ). Authors in [NM07, NM05, EA11, MZTZ10] have illustrated that embedding in AC coefficients of the 4×4 intra predicted blocks achieves relatively better visual quality.

So embedding in quantized AC coefficients of 4×4 blocks provides acceptable visual quality of the watermarked video. Authors have minimized the location map size by using the content adaptive key and designed a watermarking algorithm that can resist self collusion attack [MAAK10, NM05].

In [MZTZ10, HZC11], the authors have analyzed the relationship between the DCT coefficients and the distortion of the pixel values used in intra frame prediction to find several pairs of coefficients. In each coefficient pair, one coefficient is used for watermark embedding and the other is used to compensate the distortion drift due to embedding. However, schemes reported in the literature [NM05, EA11, XWW11, MZTZ10] are fragile in nature. All these algorithms are based on Least Significant Bit (LSB) matching or replacement technique. These watermark embedding algorithms are not robust even against common signal processing attacks, such as, salt and pepper noise, gaussian filter, etc. Moreover, the embedding framework [EA11] increases video bit rate significantly and the resulting embedding capacity is relatively less. In [NM07], a computationally expensive prediction process is required for watermark embedding and the complete decompression and re-compression of the video is required. Furthermore, the synchronization error due to embedding is not handled. Moreover, the embedding capacity is very low. Drift error propagation is not prevented in [NM07, MAAK10, NM05, EA11, XWW11]. The methods proposed in [NM07, HZC11] are non-blind in nature since the original video is required for extraction of watermark at the decoder.

The rest of the chapter is organized as follows. In the next section, the motivation of this work is described. The proposed watermarking method is described in Section 5.2. The experimental results are shown in Section 5.3. Finally, the chapter is concluded in Section 5.4.

5.1 Motivation

The above pitfalls of the existing literature motivated us to propose an I-frame based watermarking algorithm which has the following features:

- The proposed method selects the embedding regions by spatial and temporal analysis such that an acceptable visual quality can be achieved. Moreover, the watermarking method can minimize intra frame drift distortion.
- The proposed robust watermarking algorithm is blind in nature. A set of public key and private key are extracted and used for watermark detection at the decoder [MAAK10]. The public key is extracted from robust compressed domain features of the video.
- The watermarking method is made robust against self collusion attack by randomly selecting the candidate blocks using different keys for the frames, which are highly correlated.
- Watermark embedding in nonzero coefficients in selected blocks has restricted the increase in video bit rate.
- The compensation of distortion drift [MZTZ10] is performed for intra predicted blocks. However, drift error propagation to P-frames and B-frames in that GOP due to I-frame watermarking is not compensated [MZTZ10].

The main motivation of the work presented in this work is to reduce the pitfalls of the existing I-frame based video watermarking methods, such as, perceptual distortion [NM07, MAAK10], fragility [NM05, MZTZ10], non-blindness [NM07], etc. in such a way that the overall performance of the method is enhanced with respect to visual quality, robustness, blindness, bit rate increase, and security.

5.2 Proposed Method

The block diagram of the proposed embedding and extraction methods are depicted in Figure 5.1 and Figure 5.2, respectively. In this section, the selection of suitable blocks for watermarking, the candidate block selection and extraction of keys, the embedding and extraction methods, and finally the estimation of thresholds are described.

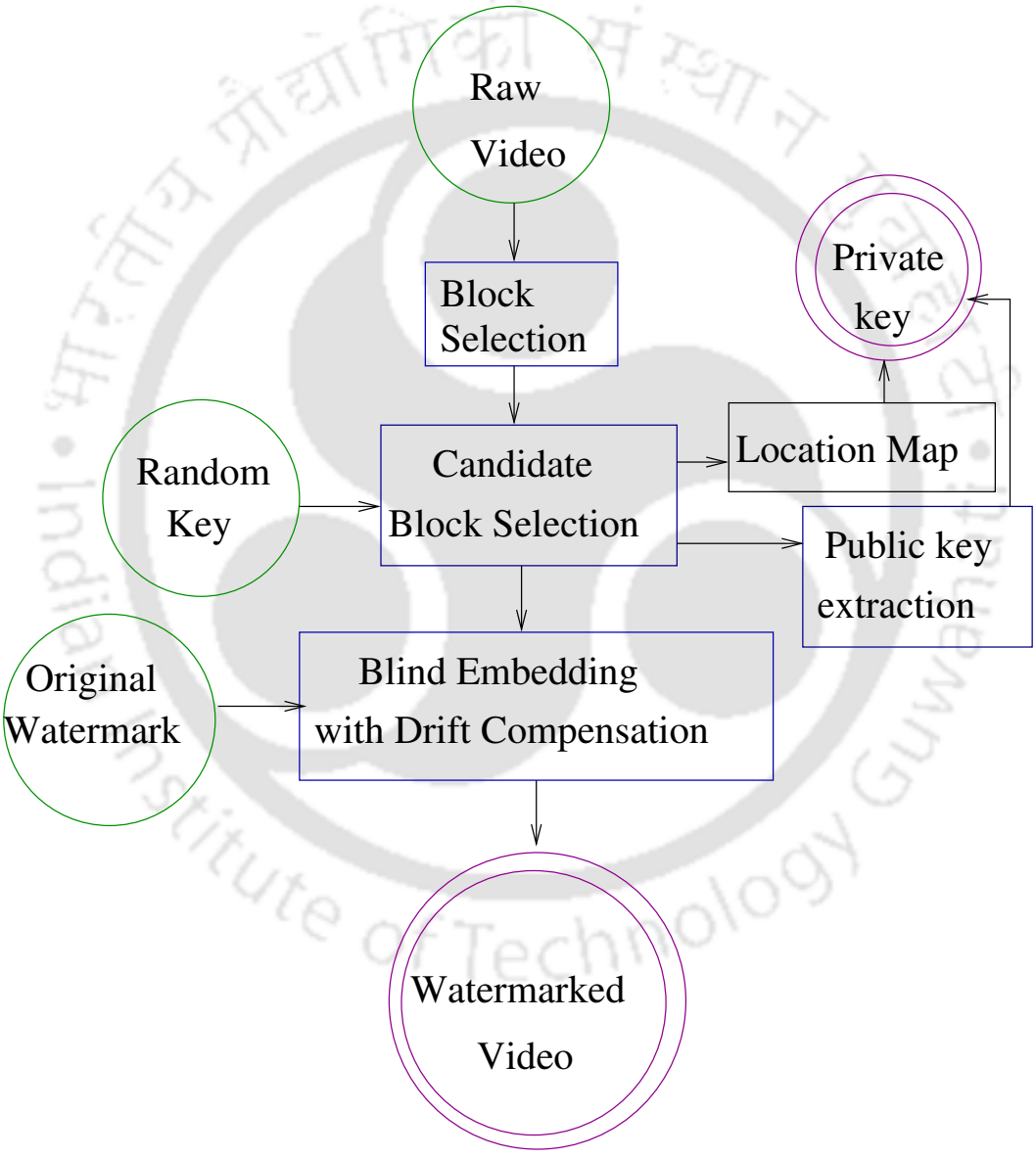


Figure 5.1 Block diagram of the proposed embedding method.

5.2.1 Block Selection

The blocks suitable for watermark embedding in the I-frame are selected, using the spatial and temporal analysis to achieve an acceptable perceptual quality of the watermarked video, less synchronization error, etc. The blocks are selected based on the following criteria:

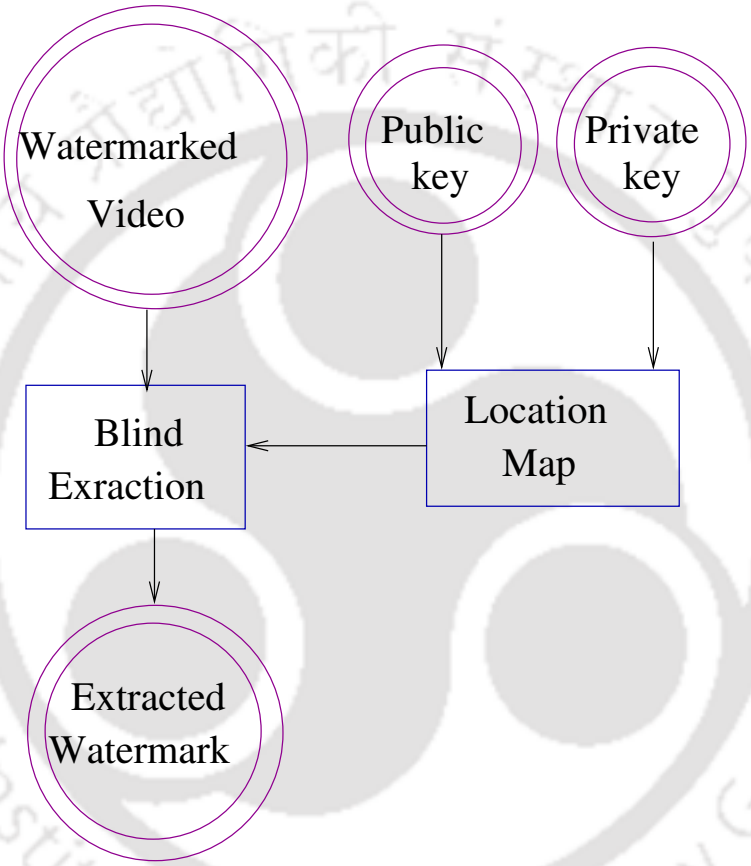


Figure 5.2 The block diagram of the proposed extraction method.

- Intra 4×4 mode blocks are selected for watermark embedding. This is due to the fact that in the I-frames, 16×16 mode blocks are chosen for the smooth regions while 4×4 mode blocks are selected for the detailed areas. Intuitively embedding in more textured blocks causes less visual artifact with respect to the human eyes and hence provide better visual quality [MAAK10] to the watermarked video.

- It is observed in [MAAK10] that the probability of changes in luminance intra

prediction modes decreases with an increase in the NNZ of the block. Therefore, 4×4 blocks having higher NNZ value is suitable for embedding. These coefficients mostly correspond to busy areas which have the capability of more embedding with less degradation. Therefore, blocks with NNZ greater than a threshold (NNZ_{th}) are selected for watermark embedding, such that,

$$NNZ > NNZ_{th} \tag{5.1}$$

where NNZ_{th} is a positive integer.

- The value of motion vectors in I-frames are zero as I-frames are intra coded. The extraction of pseudo motion vectors of I-frames based on motion vectors at the same location in the nearest P-frame of previous GOP is described in three steps as follows:

  Step 1: The nearest P-frame of previous GOP is divided into non-overlapping blocks. Motion vectors for all blocks in that P-frame is estimated.

  Step 2: In the P-frame, motion vectors of intra coded blocks are zero. Therefore, motion vector for intra coded blocks are estimated from neighboring blocks and forms a complete motion vector field [WS03].

  Step 3: Pseudo motion vector is assigned to each block in the I-frame by interpolating motion vectors at the same location in that P-frame.

  If the pseudo motion vector of a block is greater than zero, then the block is selected for watermarking.

- To minimize increase in video bit rate, only nonzero coefficients are perturbed. Making a zero coefficient to nonzero may increase the video bit rate while changing nonzero coefficients to zero may degrade the visual quality. Therefore, nonzero coefficients are also not changed to zero for watermark embedding [NM08].

- Each block should have a pair of nonzero coefficients satisfying the compensation

criterion of [MZTZ10]. The coefficients in blocks are depicted in Figure 5.3.

C_{00}	C_{10}	C_{20}	C_{30}
C_{01}	C_{11}	C_{21}	C_{31}
C_{02}	C_{12}	C_{22}	C_{32}
C_{03}	C_{13}	C_{23}	C_{33}

Figure 5.3 Different coefficient positions in a 4×4 blocks.

There are 12 coefficient pairs in two sets as follows:

$$\text{Horizontal set} = \{(C_{22}, C_{20}), (C_{02}, C_{22}), (C_{03}, C_{23}), (C_{23}, C_{03}), (C_{21}, C_{01}), (C_{01}, C_{21})\},$$

$$\text{Vertical set} = \{(C_{22}, C_{20}), (C_{20}, C_{22}), (C_{30}, C_{32}), (C_{32}, C_{30}), (C_{12}, C_{10}), (C_{10}, C_{12})\}.$$

- Unlike [MAAK10], where one watermark bit is embedded in a selected macroblock, one watermark bit is embedded in a selected 4×4 block in a macroblock. A macroblock may have more than one selected 4×4 block, which enhances the embedding capacity.

5.2.2 Candidate Block Selection and Public Key Extraction

In each I-frame, candidate blocks are selected for embedding from the set of blocks selected in the block selection process using a random key. The proposed watermarking algorithm embeds the watermark in a single quantized DCT AC coefficient of a 4×4 block. The security of the algorithm is based on the randomness of the coefficients selection process for watermark embedding. Embedding watermark in only one coefficient in a block may not cause visible artifacts. To mount an attack, the attacker has to find out the coefficient where the watermarking is done.

The selection of candidate blocks using a random key enhances the security of the proposed method. The public key and private key are generated in such a way that the the combination of keys provides the exact location of candidate blocks. The

compressed and encrypted private key is transmitted to the authorized client through secured channel and the combination of the encrypted public key and private key are used to identify the exact location of the candidate blocks for extraction of watermark at the decoder. The location map is generated based on the candidate block locations where the watermark is embedded. If the same key is used for every frame, the watermarking algorithm becomes vulnerable to the self collusion attack [NM05]. As a countermeasure, a very long key sequence is required [NM05]. Transmitting a long key, however, would make the algorithm impractical. This problem can be solved by generating the key from a combination of a public key extracted from some features of the macroblock, and a private key possessed by the copyright owner [NM05]. The public key is extracted from each macroblock and pass to a cryptographic system with the private key. Moreover, the public key should be robust to synchronization error during watermark extraction. Unlike [NM05], the public key in the proposed method is extracted from robust compressed domain features only.

In order to design a low complexity method, the compressed domain features available from H.264/AVC codec information are used for generating the public key without any further decoding. The public key would be extracted from some features of the macroblock that cannot be changed by the attacker without substantial degradation of the video quality. The HVS sensitive features of the macroblocks are used to make the public key more robust such as DC coefficients of a macroblock. If DC coefficients are used for public key extraction then the attacker could change DC coefficients by the fixed amount, which will make watermark detection impossible. In I-frames, changing a nonzero DC value to zero or zero DC value to nonzero will create significant visual artifacts [NM08] as blocks are motion compensated. Another feature is luminance intra prediction modes of a macroblock.

A 48 bits long public key (K), extracted from each macroblock based on zero or nonzero value of theDC component of a block and luminance intra prediction modes in

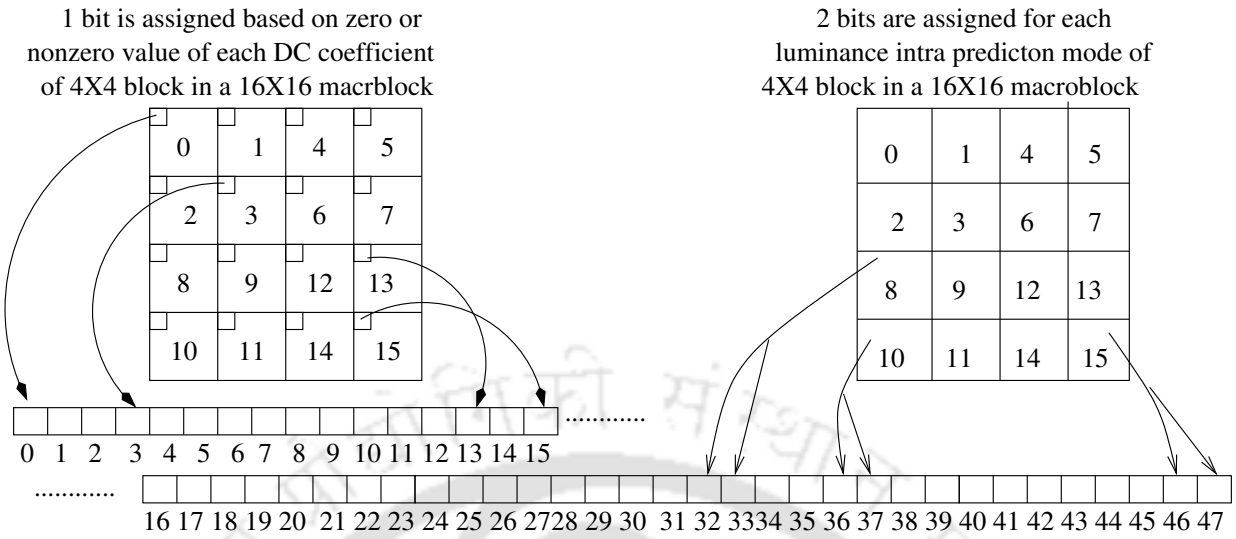


Figure 5.4 The extraction of public key from DC coefficients and luminance intra prediction modes of each macroblock [NM05].

a macroblock, is shown in Figure 5.4. The first 16 bits give each zero or nonzero value of DC coefficient of 4×4 block in a macroblock. The i^{th} bit in first 16 bits of the public key (K_i^1) is derived from i^{th} DC coefficient (DC_i) in the i^{th} 4×4 of a macroblock as follows:

$$K_i^1 = \begin{cases} 0 & DC_i = 0 \quad 0 \leq i < 16 \\ 1 & DC_i \neq 0 \quad 0 \leq i < 16. \end{cases} \tag{5.2}$$

Total nine intra prediction modes for 4×4 blocks are categorized into four groups based on prediction direction as dc mode (2), horizontal modes (1, 6, 8), vertical (0, 5, 7), and diagonal modes (3, 4). For 16×16 block size, four prediction modes are placed in four different groups *i.e.* vertical, horizontal, dc, and plane (diagonal). Since modes within a same group may be converted to each other after embedding or re-encoding, such grouping makes the public key more robust in case of alternations. Two bits for each mode is assigned. The last 32 bits of the key consists of luminance intra prediction mode information for 16 different 4×4 blocks (2 bits for each block) in a macroblock. Every two bits in the public key K_{ij}^2 denote a prediction mode in Eq. (5.3), where i and j are two consecutive bits representing the prediction mode of a 4×4 block. The blocks are scanned in zigzag scan order in a macroblock as shown in Figure 1.12(a).

The key K is made by appending K_i^1 and K_{ij}^2.

$$K_{ij}^2 = \begin{cases} 00 & \text{mode} \in \{2\} & 16 \leq i < j \leq 47 \\ 01 & \text{mode} \in \{1,6,8\} & 16 \leq i < j \leq 47 \\ 10 & \text{mode} \in \{0,5,7\} & 16 \leq i < j \leq 47 \\ 11 & \text{mode} \in \{3,4\} & 16 \leq i < j \leq 47 \end{cases} \tag{5.3}$$

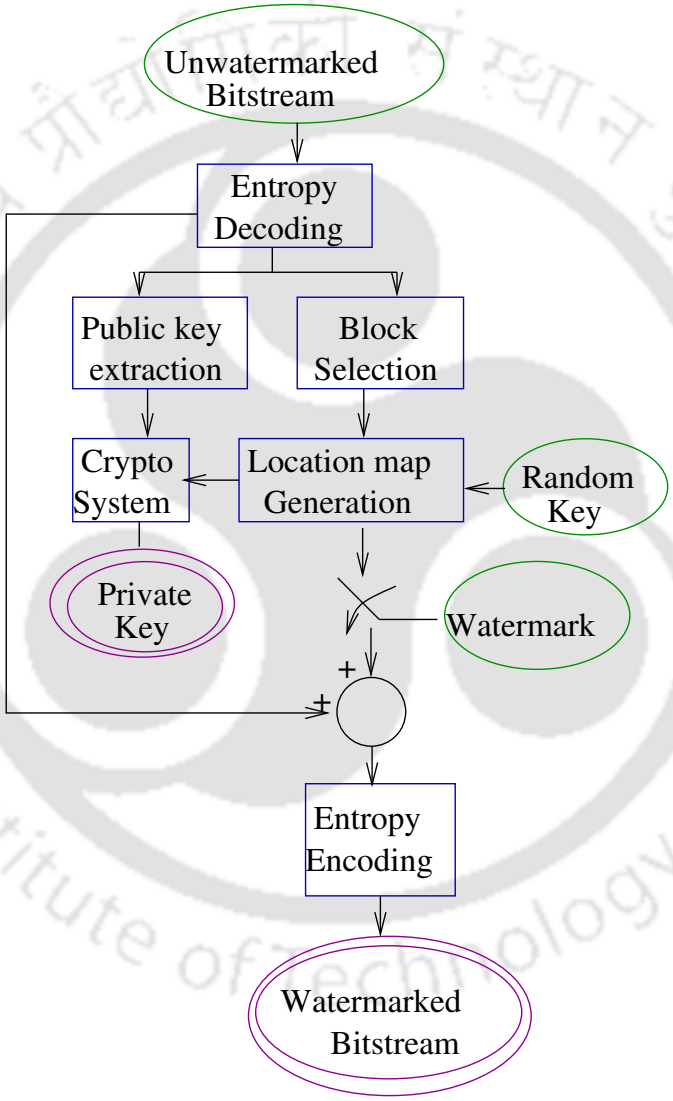


Figure 5.5 A watermark embedding framework robust to self collusion attack.

Intuitively, since the public key is content dependent, altering the public key may cause substantial degradation of the perceptual video quality and thus makes the pro-

posed method robust. The public key is scrambled using the private key to generate the resultant key which specifies the candidate block locations used for embedding. The public key is encrypted using modulo 2 arithmetic. The encrypted key is compressed losslessly using run length coding and is available to all clients.

The private key is estimated using the public key and the generated location map at the decoder side. The private key is compressed using run length coding and sent to the authorized client through secure channel. The size of the compressed private key is small. In most of the literature, the location map is sent to the decoder through secured channels. In the proposed method, only a compressed private key is sent securely to the authorized client. Alike [NM05], the proposed watermarking algorithm can resist self collusion attack as shown in Figure 5.5 by selecting different keys for similar frames in a video.

A location map [MAAK10] is a file that contain candidate block locations of the encoder. It is used for watermark extraction at the decoder. In the proposed algorithm, a location map is generated from the combination of the private key and public key at the decoder.

5.2.3 Embedding and Extraction

A bipolar watermark is embedded invisibly in AC coefficients of a 4×4 block in I-frames. To minimize synchronization error, only 4×4 blocks having NNZ greater than the threshold NNZ_{th} are used for embedding in Section 5.2.1 [MAAK10]. The threshold R_T is also used to enhance the robustness of the proposed method, where R_T is a positive integer.

Ma *et. al.* in [MZTZ10] have described a simple drift signal compensation method for watermark embedding in AC coefficients of intra predicted blocks. The drift signal compensation method described in [MZTZ10] is that due to watermark embedding if any small integer Δ is added to an AC coefficient then the embedding distortion is

accumulated in the middle of two rows in the 4×4 block. So Δ will be subtracted from the paired coefficient selected from the rows in the middle. Thus subtracting Δ to paired coefficient will nullify the drift error propagation. The matrix (M) shows how drift error is compensated for watermark embedding in AC coefficients in a 4×4 block.

$$M = \begin{pmatrix} 0 & 0 & 0 & \Delta \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\Delta \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

The watermark is embedded using the embedding algorithm, which is described in Algorithm 6. The embedding method is blind in nature, where original (unwatermarked) video is not required at the decoder for extraction of watermark. A location map is generated to save the candidate block location and send to the decoder. In the coefficient pair, nonzero AC coefficients of a candidate block, denoted by AC_x and AC_y, respectively, where $|AC_x| > 0$ and $|AC_y| > 0$ are selected. The embedding rule is proposed as follows:

$$\begin{aligned} &\textbf{If} \text{ watermark bit is } 0 \ \ \textbf{then } |AC_x| > |AC_y| \\ &\textbf{If} \text{ watermark bit is } 1 \ \textbf{then } |AC_x| < |AC_y| \end{aligned} \tag{5.4}$$

The difference between the coefficient pairs and the sign of the difference, denoted by $diff_{xy}$ and $sign(diff_{xy})$, are defined as follows:

$$diff_{xy} = \begin{cases} \left\lfloor \frac{AC_y - AC_x}{2} \right\rfloor & \text{if } |AC_x| < |AC_y| \\ \left\lfloor \frac{AC_x - AC_y}{2} \right\rfloor & \text{if } |AC_x| > |AC_y| \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

and

$$sign(diff_{xy}) = \begin{cases} positive\ (+1) & \text{if } diff_{xy} \geq 0 \\ negative\ (-1) & \text{otherwise,} \end{cases} \tag{5.6}$$

respectively. If watermark bit is zero (0) and $|AC_x|$ is less than $|AC_y|$ then the modified coefficients, denoted by AC'_x and AC'_y, are given by

$$AC'_x = AC_x + diff_{xy} + sign(diff_{xy}) \times R_T$$

and

$$AC'_y = AC_y - diff_{xy} - sign(diff_{xy}) \times R_T,$$

respectively, where threshold R_T is added to keep significant difference between the coefficients that increases the robustness of the proposed method. The sign of R_T depends on $diff_{xy}$. Similarly, if watermark bit is unity (1) and $|AC_y|$ is less than $|AC_x|$ then

$$AC'_y = AC_y + diff_{xy} + sign(diff_{xy}) \times R_T$$

and

$$AC'_x = AC_x - diff_{xy} - sign(diff_{xy}) \times R_T.$$

If watermark bit is 0 and $|AC_x|$ greater than $|AC_y|$ or watermark bit is 1 and $|AC_x|$ less than $|AC_y|$, no changes will be made. The embedding algorithm for each candidate block is given in Algorithm 6. If $diff_{xy}$ is large, then perceptual distortions may occur. On the other hand, the selection of coefficient pair with small $diff_{xy}$ reduces the embedding capacity. So a trade-off exists between the visual quality and embedding capacity. The selection of blocks suitable for watermark embedding can be further restricted by incorporating a visual quality threshold, denoted by V_T. The maximum value of $diff_{xy}$ can be limited by the visual quality threshold (V_T) such that

$$|diff_{xy}| \leq V_T. \tag{5.7}$$

The watermark extraction is performed at the decoder after entropy decoding. The extraction procedure is the reverse process of watermark embedding. The block locations where the watermark is embedded are saved in a location map during the

Algorithm 6: Embedding Algorithm

Input: Candidate blocks in I-frames

Output: Watermarked blocks

for each candidate block in I-frames **loop**

Select AC_x and AC_y, two nonzero AC coefficients in a coefficient pair

if (watermark bit is 0) **and** ($|AC_x| \leq |AC_y|$) **then**

$diff_{xy} = \left\lfloor \frac{AC_y - AC_x}{2} \right\rfloor$

$AC'_x = AC_x + diff_{xy} + sign(diff_{xy}) \times R_T$

$AC'_y = AC_y - diff_{xy} - sign(diff_{xy}) \times R_T$

elseif (watermark bit is 1) **and** ($|AC_x| \geq |AC_y|$) **then**

$diff_{xy} = \left\lfloor \frac{AC_x - AC_y}{2} \right\rfloor$

$AC'_y = AC_y + diff_{xy} + sign(diff_{xy}) \times R_T$

$AC'_x = AC_x - diff_{xy} - sign(diff_{xy}) \times R_T$

else

no change in coefficients

end

end

embedding process. This location map is used by the decoder during the extraction process to get candidate blocks. The embedded watermark bit is extracted from every candidate blocks in watermarked video as follows:

$$\text{The watermark bit} = \begin{cases} 0 & |AC'_x| > |AC'_y| \\ 1 & \text{otherwise,} \end{cases}$$

where AC'_x and AC'_y are nonzero coefficients in a coefficient pair of a block in a I-frame of the watermarked video.

Since the proposed watermarking method does not depend on the LSB or sign of the coefficients, it is robust against any transformations (like blurring or geometric transformation) and lossy compressions (like H.264) can easily destroy LSBs of coefficients [SP96]. The robustness are enhanced by incorporating the robustness threshold (R_T). The visual quality of the watermarked video is acceptable. The drift error compensation and selection of suitable blocks based on spatial and temporal characteristics of the video help to minimize embedding distortion. In addition, V_T is used to further enhance the perceptual quality of the watermarked video.

Example 6 *Assume, the values of two nonzero coefficients in coefficients paired coefficients (AC_x, AC_y) and R_T in a block are* 3, 6, *and* 4, *respectively.*

Case 1: The watermark bit is 0.

If watermark bit is 0 *then $|AC_x|$ should be greater than $|AC_y|$ according to the embedding rule in Eq. (5.4). Since $|AC_y| > |AC_x|$, the difference between AC_x and AC_y will be*

$$AC_y - AC_x = 3$$
$$diff_{xy} = \lfloor \tfrac{3}{2} \rfloor = 1$$
$$sign(diff_{xy}) = +1.$$

So AC_x and AC_y are modified to AC'_x and AC'_y, respectively as follows:

$$AC'_x = AC_x + 1 + 4 = 8$$

$$AC'_y = AC'_y - 1 - 4 = 1.$$

During the extraction process at the decoder, if $|AC'_x| > |AC'_y|$ *then the extracted watermark bit is* 0.

Case 2: The watermark bit is 1.

If watermark bit is 1 *then no coefficients will be changed as it follows the embedding rule of Eq. (5.4) and value of the watermarked coefficients will be*

$$AC_x = AC'_x = 3 \text{ and } AC_y = AC'_y = 6.$$

During the extraction process at the decoder, if $|AC'_x| < |AC'_y|$ *then the extracted watermark bit is* 1.

Example 7 *Assume, the values of two nonzero coefficients in coefficients paired coefficients* (AC_x, AC_y) *and* R_T *in a block are* 3, -6, *and* 4, *respectively.*

Case 1: The watermark bit is 0.

If watermark bit is 0 *then* $|AC_x|$ *should be greater than* $|AC_y|$ *according to the embedding rule in Eq. (5.4). Since* $|AC_y| > |AC_x|$, *the difference between* AC_x *and* AC_y *will be*

$$AC_y - AC_x = -9$$
$$diff_{xy} = \lfloor \tfrac{-9}{2} \rfloor = -4$$
$$sign(diff_{xy}) = -1.$$

So AC_x *and* AC_y *are modified to* AC'_x *and* AC'_y, *respectively as follows:*

$$AC'_x = AC_x - 4 - 4 = -5$$
$$AC'_y = AC'_y + 4 + 4 = 2.$$

During the extraction process at the decoder, if $|AC'_x| > |AC'_y|$ *then the extracted watermark bit is* 0.

Case 2: The watermark bit is 1.

If watermark bit is 1 *then no coefficients will be changed as it follows the embedding rule of Eq. (5.4) and value of the watermarked coefficients will be*

$$AC_x = AC'_x = 3 \text{ and } AC_y = AC'_y = -6.$$

During the extraction process at the decoder, if $|AC'_x| < |AC'_y|$ *then the extracted watermark bit is* 1.

5.2.4 Threshold Selection

In the proposed watermark embedding algorithm, the visual quality threshold (V_T) is used to control the degradation in visual quality of the watermarked video while robustness threshold (R_T) is used to increase robustness of the proposed method against different attacks.

Robustness Threshold (R_T): The perceptual quality of the watermarked video and robustness of the proposed watermarking method are measured using PSNR and BER [refer to Appendix] (against recompression error) with varying R_T is depicted in Table 5.1. It is observed that if R_T increases, robustness increases, but visual quality degrades. Similarly, if R_T decreases, robustness decrease, but the watermarked video will have better visual quality. Therefore, a trade-off exists between visual quality and robustness while selecting the value of R_T based on the exhaustive experimental results (refer to Table 5.1).

Visual Quality Threshold (V_T): The change in perceptual quality of the watermarked video (using PSNR) and the embedding capacity of the proposed watermarking method [refer to Appendix] with varying V_T is depicted in Table 5.2. It is observed that if V_T increases, the embedding capacity increases, but visual quality degrades. Similarly, if V_T decreases, embedding capacity decrease, but the watermarked video will have better visual quality. Therefore, a trade-off is considered between visual quality and embedding capacity while selecting the value of V_T. In addition, with the increase in the value of V_T, bit increase rate (BIR) will also increase.

Table 5.1 Selection of robustness threshold R_T based on PSNR and BER, when $V_T = 10$.

Video Sequence	R_T	Average PSNR	Average Robustness
Carphone	1	37.37	80
	4	36.41	85
	7	35.79	88
Foreman	1	35.78	81
	4	35.37	84
	7	34.29	86
News	**1**	37.52	84
	4	37.19	85
	7	36.41	87
Salesman	1	36.45	78
	4	36.08	82
	7	35.15	83
Suzie	1	36.22	79
	4	35.59	86
	7	34.78	89
Trevor	1	36.38	78
	4	35.85	81
	7	35.51	85

Table 5.2 Selection of visual quality threshold V_T based on PSNR and embedding capacity, when $R_T = 4$.

Video Sequence	V_T	Average PSNR	Average Embedding Capacity
Carphone	6	37.79	0.45
	10	36.81	0.56
	14	35.87	0.68
Foreman	6	36.08	0.58
	10	34.67	0.74
	14	34.09	0.80
News	**6**	37.82	0.45
	10	37.19	0.50
	14	36.52	0.58
Salesman	6	36.25	0.40
	10	35.58	0.49
	14	34.85	0.53
Suzie	6	36.32	0.52
	10	35.45	0.58
	14	34.96	0.67
Trevor	6	37.03	0.48
	10	36.15	0.55
	14	35.31	0.59

5.3 Experimental Results

The proposed method is implemented using H.264/AVC [Ric10] reference software JM 17.2 [S08]. The experimental setup is presented in Table 5.3.

Table 5.3 Experimental Setup

Parameters	**Values**
Video Format	Quarter Common Intermediate format (QCIF)
Frame Resolution	176×144
Frame Rate	30 frames per second
Codec Used	H.264/AVC reference software JM 17.2
Intra Period	13
GOP Structure	IBBBPBBBPBBBP
Encoding Profile	High Profile
Entropy Encoding	CAVLC
Number of frames encoded	95 frames per video
Quantization Parameter (QP)	28
Payload	{100, 200, 300, 400, 500}
Threshold R_T	4
Threshold V_T	10
Threshold NNZ_{th}	5
Video Sequences	Foreman, Carphone, News, Salesman, Suzie, Trevor

The value of NNZ_{th} is considered as 5 based on the existing scheme in [MAAK10] and the value of R_T is taken as 4 based on the experimental results tabulated in Table 5.1. The simulation results for the embedding capacity, visual quality of the watermarked video, increase in video bit rate, and robustness against different attacks are shown in Section 5.3.1, Section 5.3.2, and Section 5.3.3, respectively.

5.3.1 Embedding Capacity

The embedding capacity of the proposed method depends on the block selection process (refer to Section 5.2.1) such as NNZ and motion vector. Blocks that are selected using the block selection criteria in first 10 GOP of different videos are shown in Figure 5.6. It is observed from the results that most of the videos have an acceptable embedding capacity. The rate of blocks selected for watermarking using the block selection process

(refer to Section 5.2.1) per GOP verses GOP number of each videos are mapped in Figure 5.6.

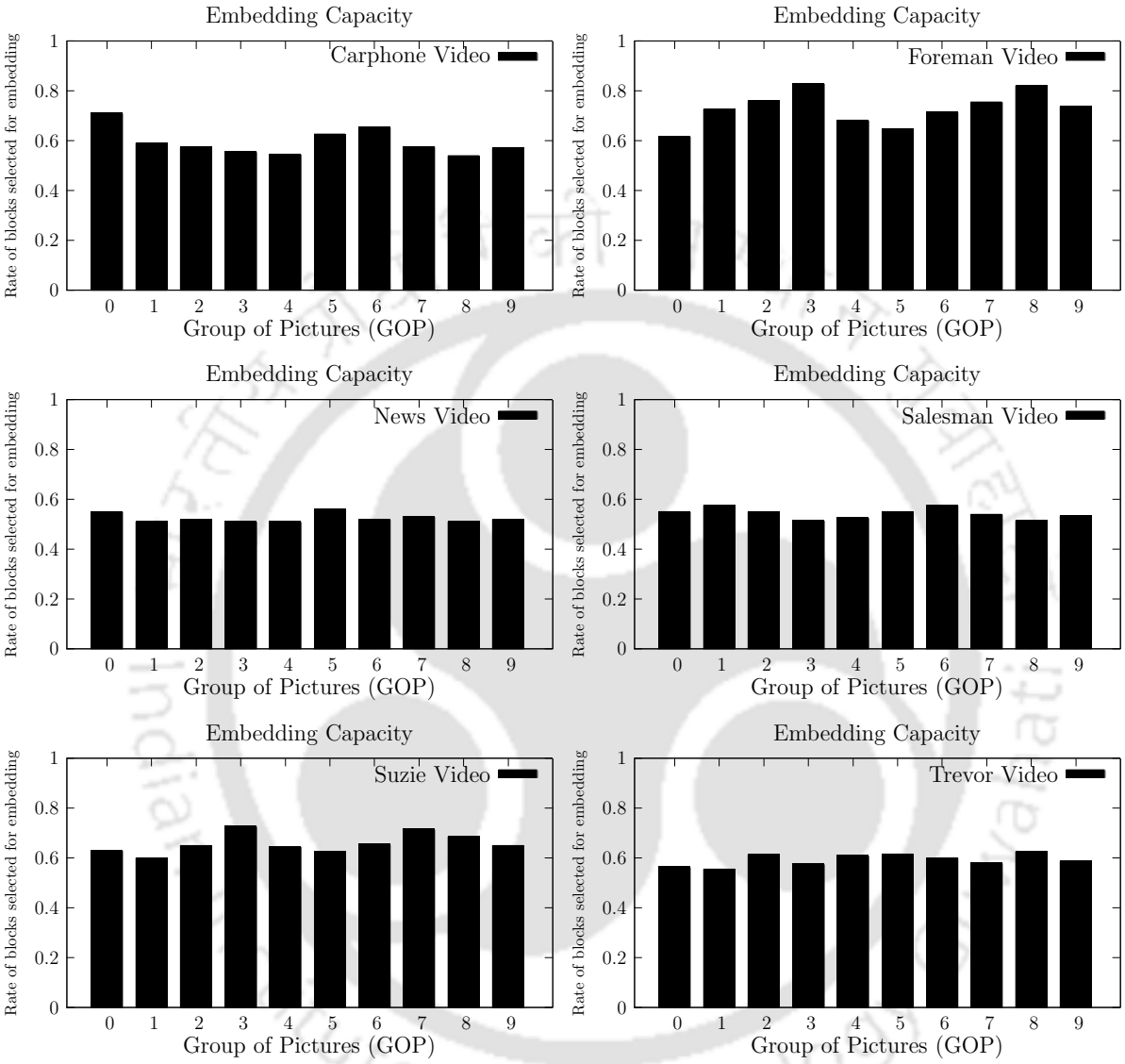


Figure 5.6 Embedding capacity based on block selection for Carphone, Foreman, News video, Salesman, Suzie, and Trevor video.

5.3.2 Visual Quality and Bit Increase Rate

The invisible watermark sequence is embedded using the proposed embedding algorithm (Algorithm 6). The visual quality of watermarked video is compared with recent existing methods [MAAK10, MZTZ10, NM07] based on Video Quality Metric (VQM)

and Peak Signal-to-Noise Ratio (PSNR). Bit Increase Rate (BIR) is also calculated and compared with existing methods [MAAK10, MZTZ10, NM07].

VQM, PSNR, and BIR of the proposed method at payload={100, 150, 200, 250, 300} for an average of 95 frames for video sequences, such as, {Carphone, Foreman, News, Salesman, Suzie, Trevor} are depicted in Table 5.4. VQM in Table 5.4 is in the order of 10^{-2}. The results of VQM and PSNR indicate that watermarked videos have acceptable visual quality. In case of BIR, the increase in video bit rate is in the order of 10^{-3} which can be regarded as nominal. In the proposed method, the magnitude value of only AC coefficients of a block are perturbed without disturbing DC value and the NNZ is kept same for a block, which intuitively avoid the high BIR increase.

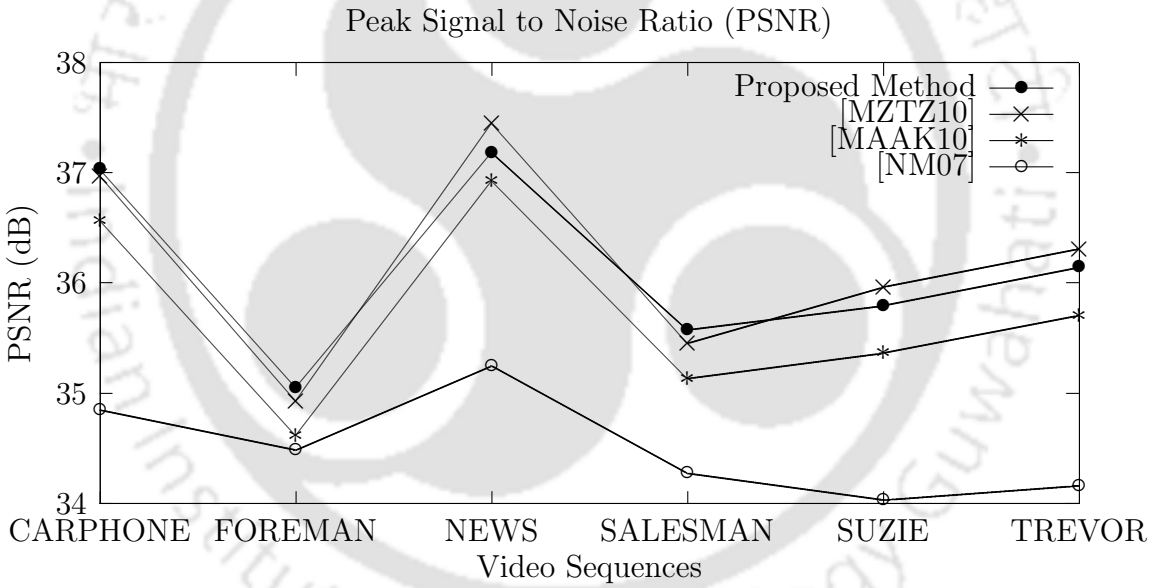


Figure 5.7 Average Peak Signal-to-Noise Ratio (PSNR).

The comparison results of the proposed method with the existing methods [MAAK10, MZTZ10, NM07] with respect to the average PSNR, average VQM, and average BIR, is presented in Figure 5.7, Figure 5.8, and Figure 5.9, respectively. BIR (Figure 5.9) of the proposed method shows a marginal increase in video bit rate. Alike [MAAK10], in some scenario the BIR decreases with the increase of payload. It is observed in Figure 5.7 that the visual quality (PSNR) of the watermarked video is comparable with the

Table 5.4 Results for PSNR, VQM, and BIR of the proposed method.

Sequence	Payload	PSNR (dB)	VQM$\times 10^{-2}$	BIR$\times 10^{-3}$
Carphone	100	37.87	4.44	0.55
	200	37.31	4.48	1.03
	300	37.01	4.63	0.6
	400	36.66	5.32	0.98
	500	36.29	6.12	0.53
Foreman	100	36.11	3.05	0.69
	200	35.88	3.57	0.77
	300	34.63	3.46	0.38
	400	34.42	4.21	1.8
	500	34.21	4.33	1.01
News	100	37.74	3.22	1.08
	200	37.43	3.33	0.55
	300	37.1	3.65	0.17
	400	36.92	3.94	0.49
	500	36.71	4.13	0.71
Salesman	100	36.54	4.51	0.54
	200	36.01	4.66	1.3
	300	35.45	4.77	1.1
	400	35.11	4.89	1.4
	500	34.77	5.29	0.68
Suzie	100	36.93	3.42	3.9
	200	36.01	3.48	1.2
	300	35.65	3.57	0.47
	400	35.35	3.81	0.52
	500	35.04	39.8	0.99
Trevor	100	36.99	2.62	0.76
	200	36.62	2.78	0.03
	300	36.38	3.25	1.3
	400	35.5	3.45	0.92
	500	35.21	3.87	0.83

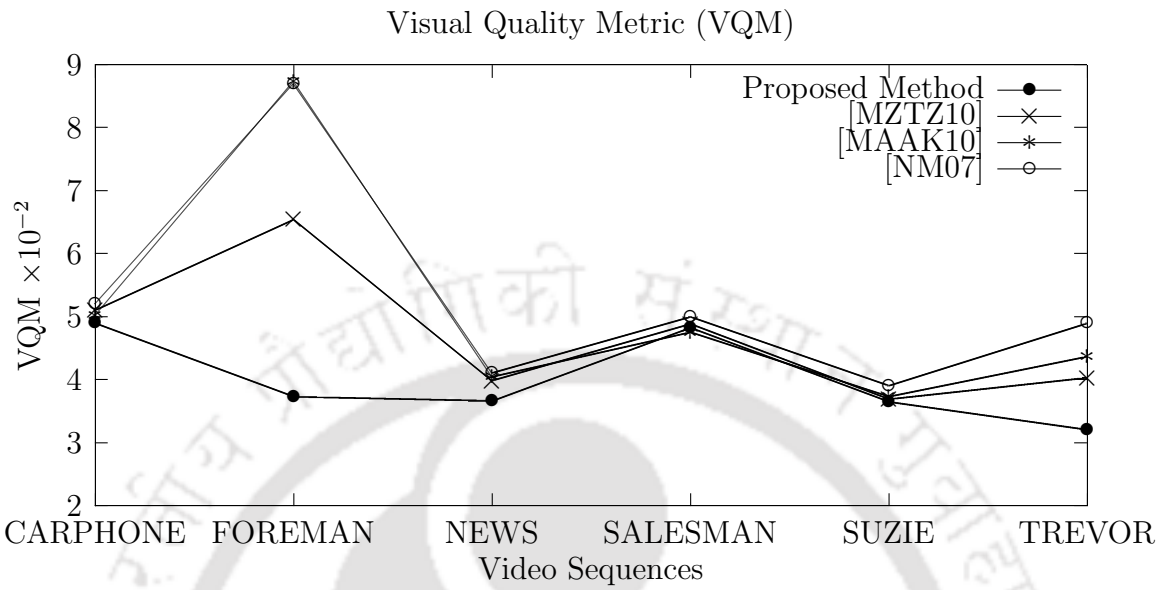


Figure 5.8 Average Visual Quality Metric (VQM).

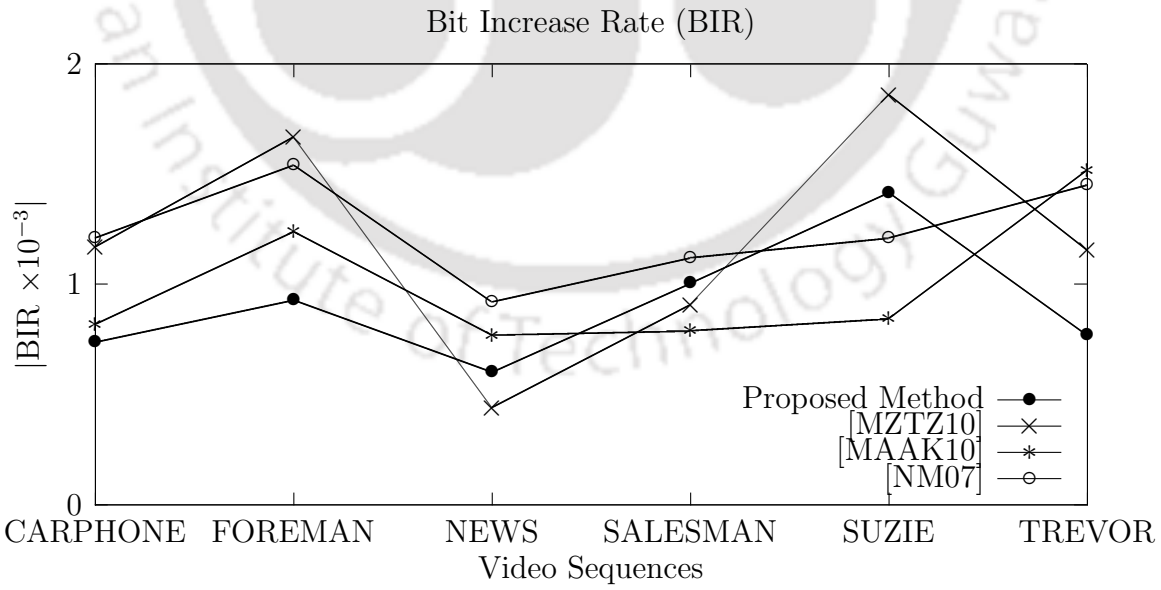


Figure 5.9 Average Bit Increase Rate (BIR).

method proposed in [MZTZ10] and outperforms methods [MAAK10, NM07]. Figure 5.8 also depicts that the proposed method has acceptable perceptual quality.

5.3.3 Robustness to Attacks

In this subsection, the bit error rate is used for evaluating the robustness [refer to Appendix] of the proposed method against different attacks. The number of error bits is determined over all frames of watermarked video stream. The information about the embedding locations is saved as a location map during the embedding process. This location map is communicated to the decoder during the extraction process. In the absence of location map, if a watermarked location is not detected or an unwatermarked position is selected incorrectly, the synchronization in the watermark sequence will be lost. This decreases the robustness of the watermarking method. Figure 5.10 shows the robustness of the proposed method against varying QP from 28 to range of 20 to 36, filtering using gaussian and circular averaging filter, and and in presence of salt and pepper noise for the *foreman* video.

The comparison results for the proposed method (with location map) with the existing methods [MAAK10, MZTZ10] with respect to the recompression error is depicted in Figure 5.11. Similar results for varying QP [from 28 to 30] and [from 28 to 26] are presented in Figure 5.12 and Figure 5.13 respectively. In [MAAK10], the NNZ in a block is decreased to embed a watermark bit. In [MZTZ10], the watermark is extracted based on LSB of the embedding coefficients.

In the proposed method, the absolute value of nonzero coefficient is changed in a block and the watermark is extracted from the absolute difference between the selected coefficients of two sequences. Intuitively, this may be the reason for the proposed method to perform better against re-encoding (QP 28 to 30 and 26) and re-compression error.

Figure 5.14, Figure 5.15, Figure 5.16, and Figure 5.17 illustrate the average robust-

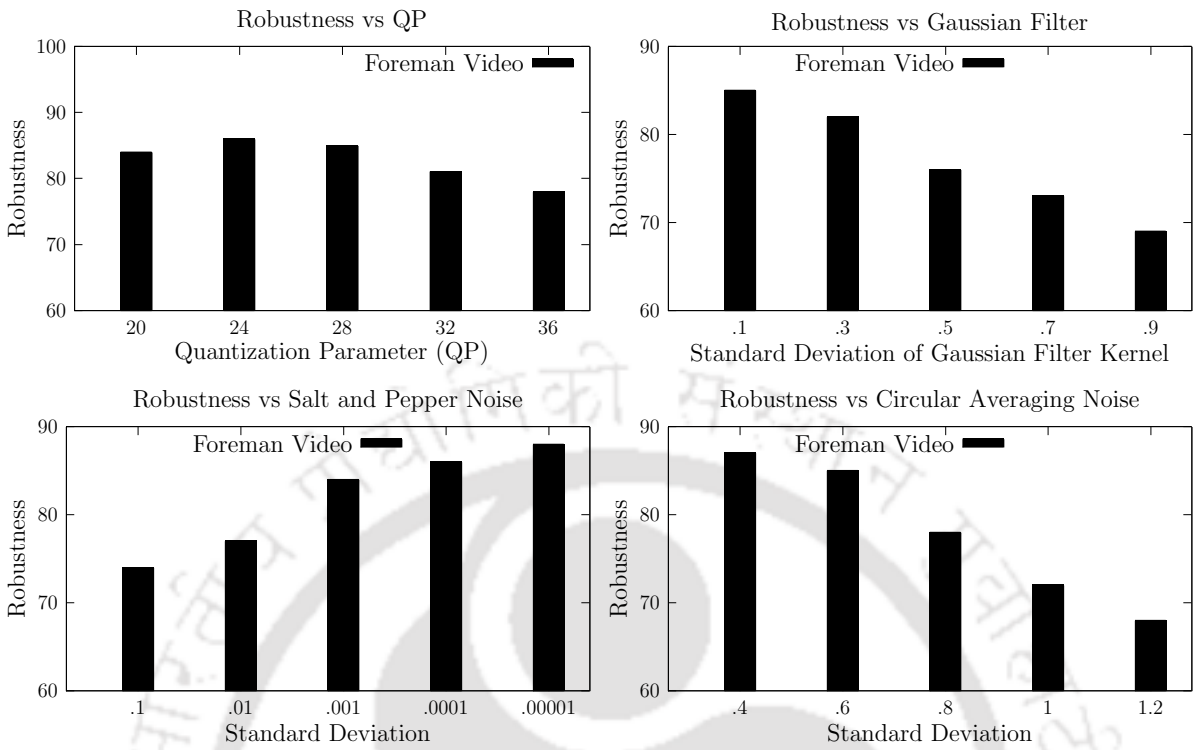


Figure 5.10 The change in robustness with QP, gaussian filter, salt and pepper noise, and circular averaging filter in the foreman video.

ness against salt and pepper noise, circular averaging filter, additive white gaussian noise, and gaussian filter, respectively providing location map to the decoder for the proposed method with [MAAK10] and [MZTZ10], where gaussian noise density = 0.001, salt and pepper noise density = 0.001, circular averaging filter r = 0.06, gaussian filter [5×5], and sigma=0.3 [MAAK10]. From these results, it can be conclude that proposed method outperforms other existing methods [MAAK10, MZTZ10].

Figure 5.18, Figure 5.19, Figure 5.20, Figure 5.21, Figure 5.22, Figure 5.23, and Figure 5.24 depict the average robustness against re-compression error, "changing quantization parameter (QP)" from 28 to 30 and 26, salt and pepper noise, circular averaging filter, additive white gaussian noise, and gaussian filter, respectively without using location map to the decoder for the proposed method with [MAAK10] and [MZTZ10], where gaussian noise density = 0.001, salt and pepper noise density = 0.001, circular averaging filter r = 0.06, gaussian filter [5×5], and sigma=0.3 [MAAK10]. It is observed from

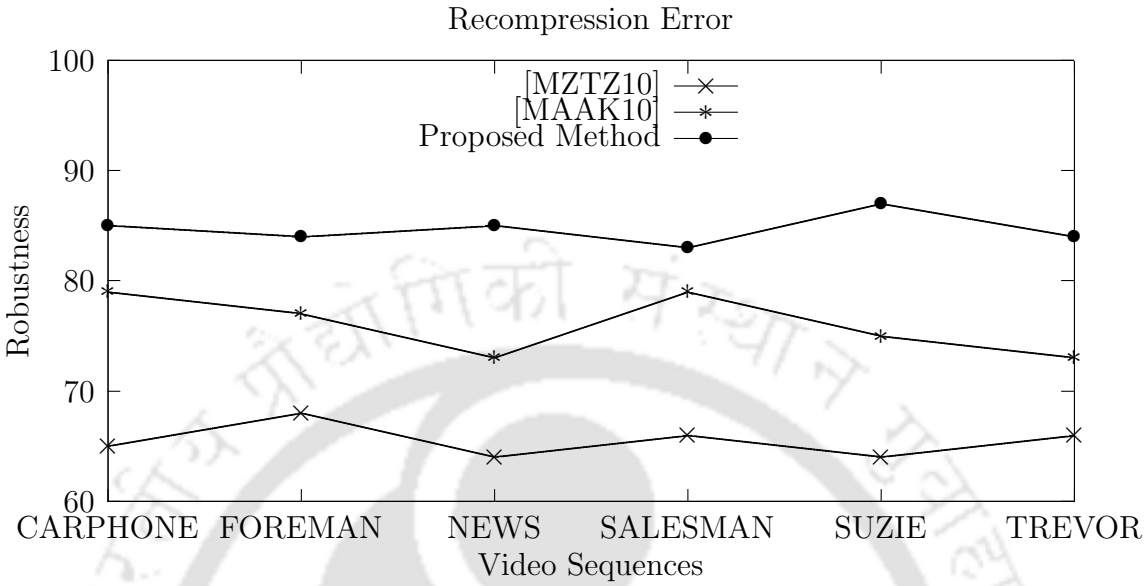


Figure 5.11 Average robustness against recompression error.

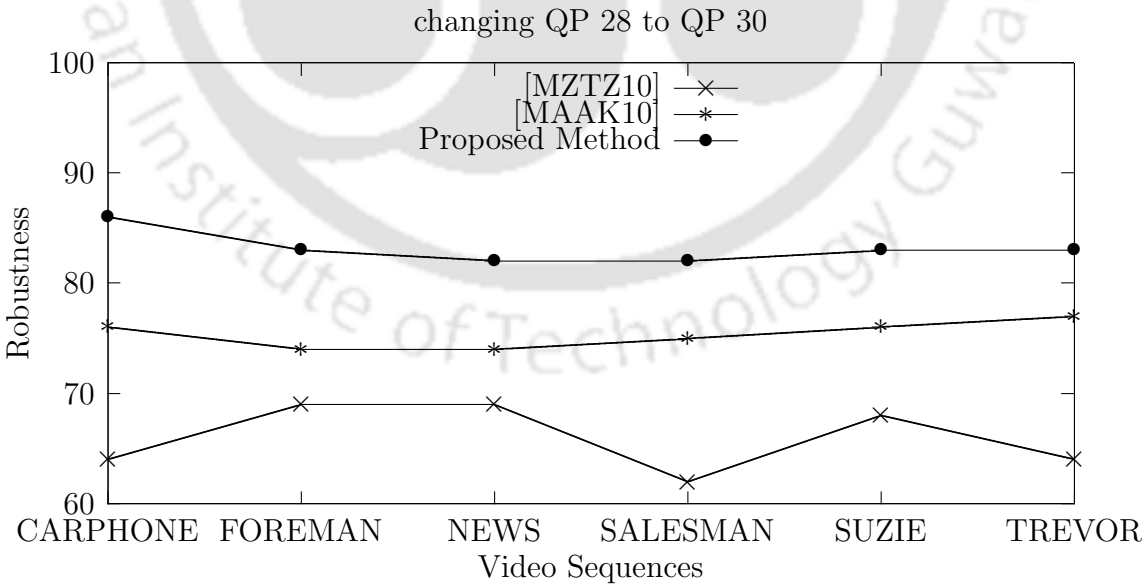


Figure 5.12 Average robustness against changing QP 28 to QP 30.

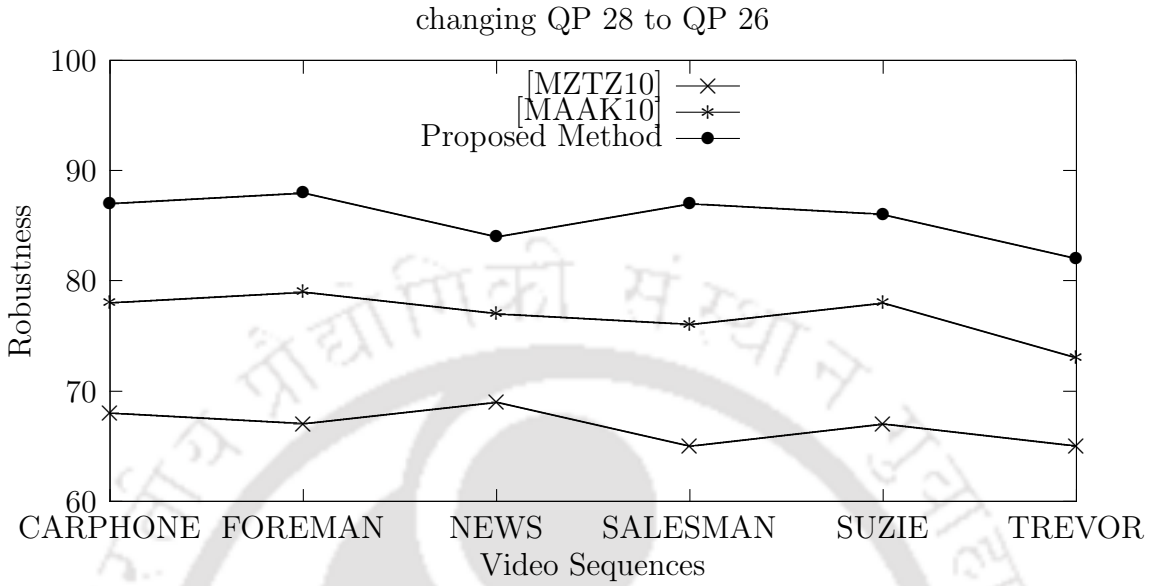


Figure 5.13 Average robustness against changing QP 28 to QP 30.

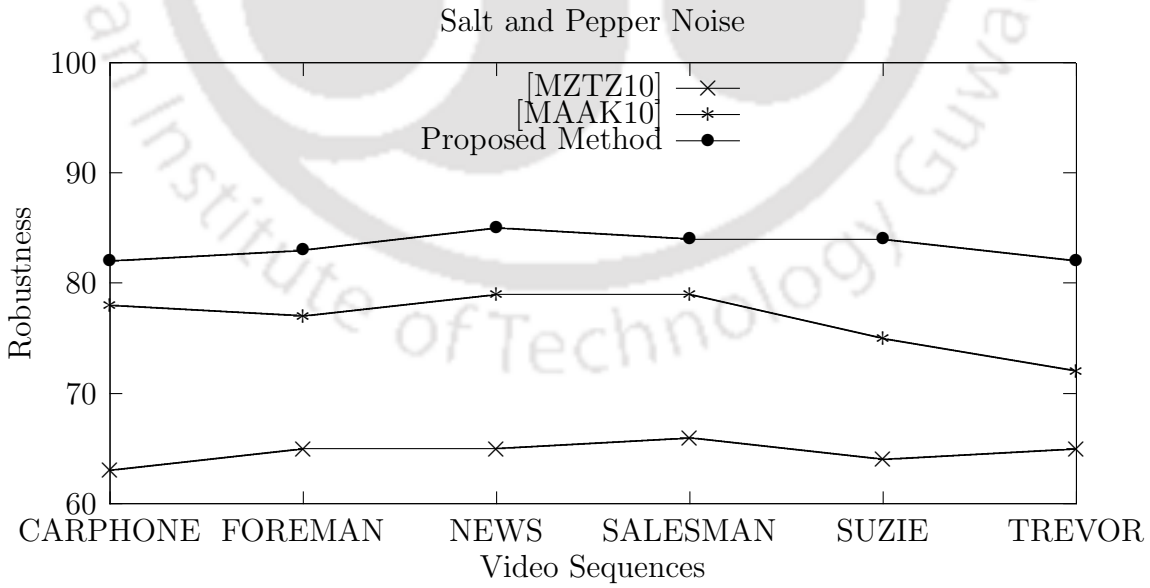


Figure 5.14 Average robustness against salt and pepper noise.

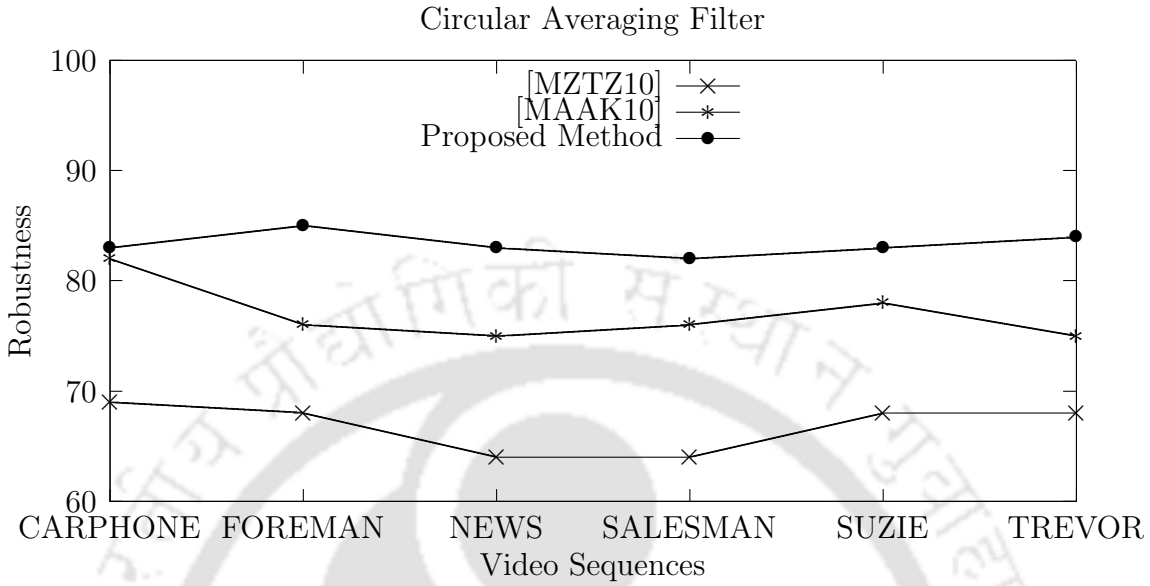


Figure 5.15 Average robustness against circular averaging filter.

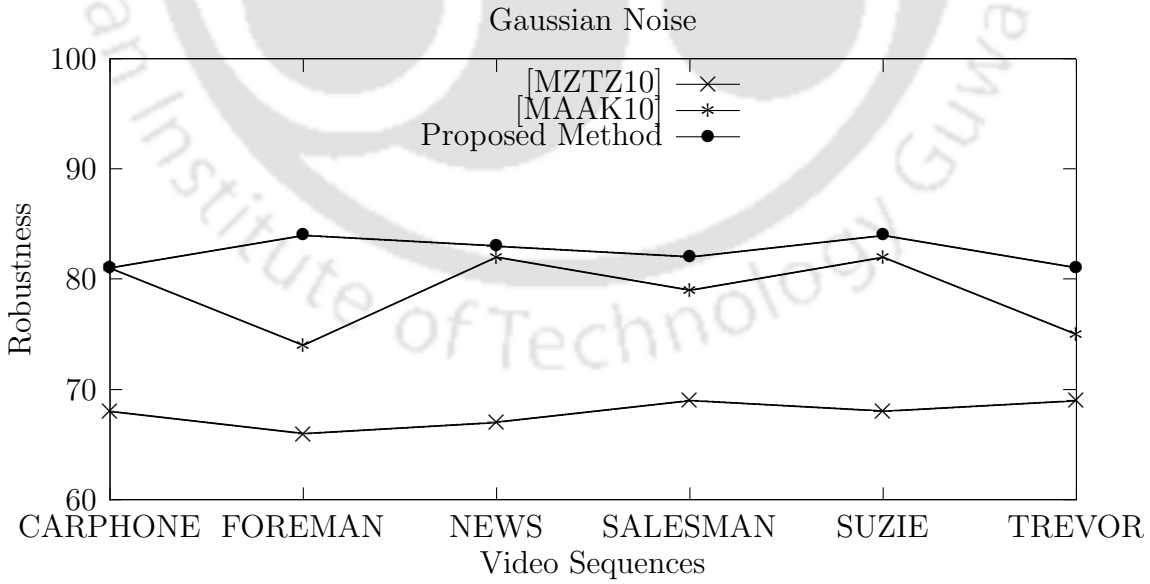


Figure 5.16 Average robustness against gaussian noise.

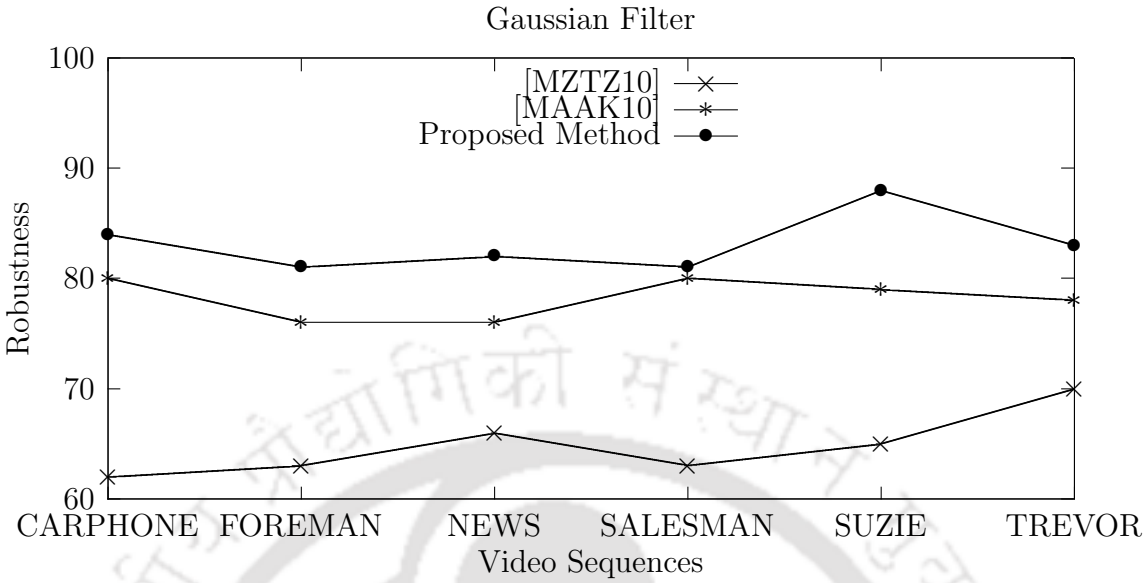


Figure 5.17 Average robustness against gaussian filter.

these results that proposed method outperforms existing methods [MAAK10, MZTZ10] in most of the cases.

5.4 Summary

In this chapter, a robust watermark embedding algorithm for I-frames having better visual quality and acceptable bit rate is designed. Intra frame drift error propagation is prevented. The hybrid algorithm uses the advantages of the existing literature and removes the pitfalls. The embedding capacity is enhanced by embedding watermark bits in 4×4 selected blocks. Both location aware and unaware detection of watermark bits at the decoder is performed and the robustness are compared with state of the art literature. The public key and private keys are extracted to reduce the size of location map and to resist self collusion attack.

Motion-compensated-temporal-frame averaging (MC-TFA) attack has become a popular attack, especially when watermarking is performed in frame-by-frame manner [PDB09]. Intuitively, prevention of distortion drift in the proposed method helps

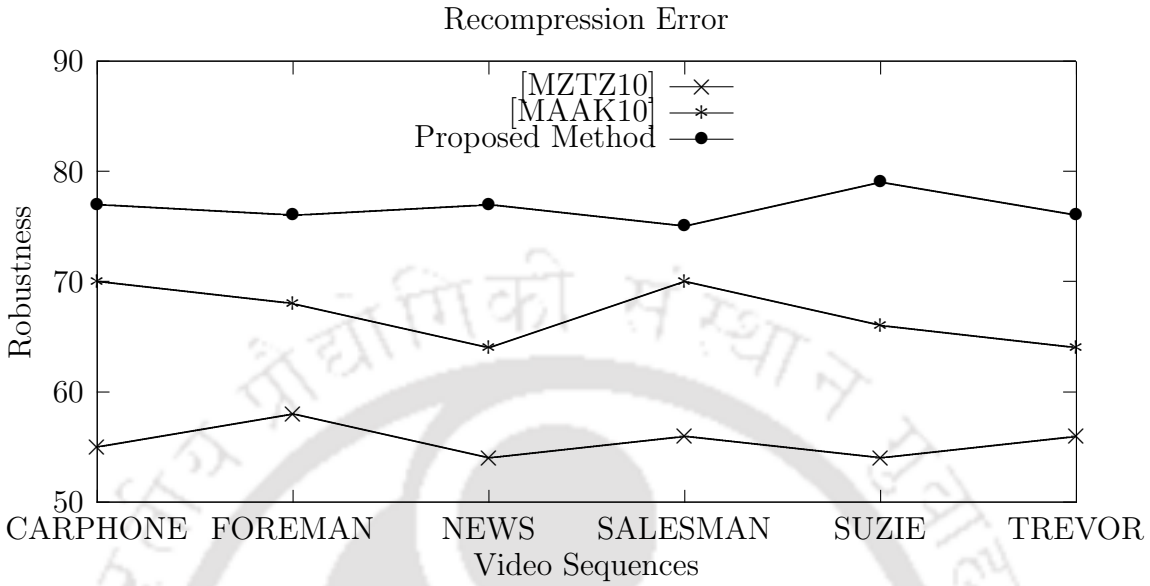


Figure 5.18 Average robustness against recompression error without using location map.

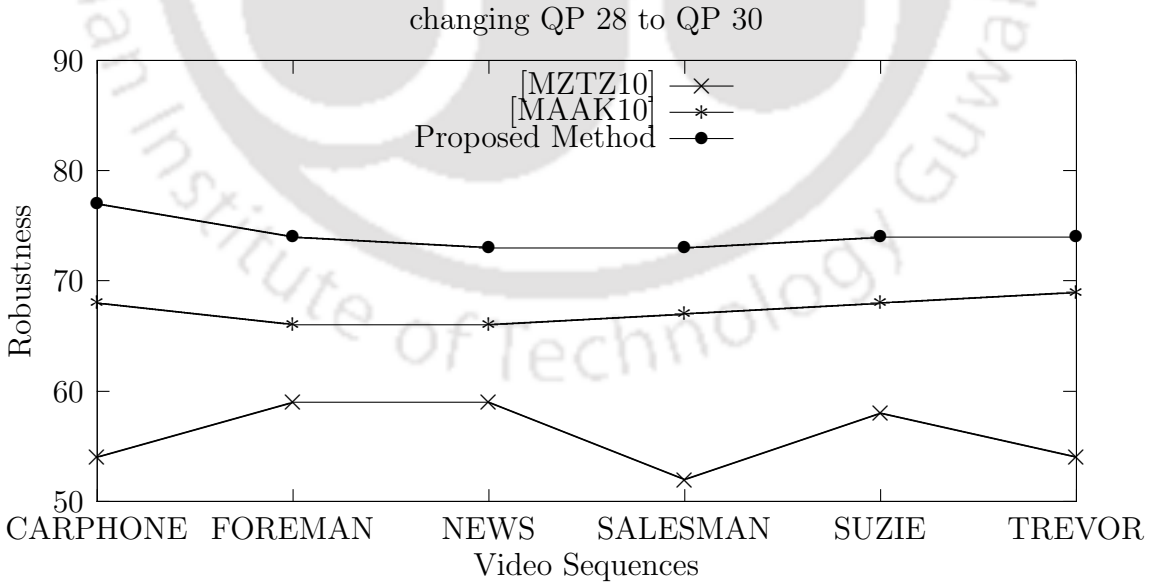


Figure 5.19 Average robustness against changing QP 28 to QP 30 without using location map.

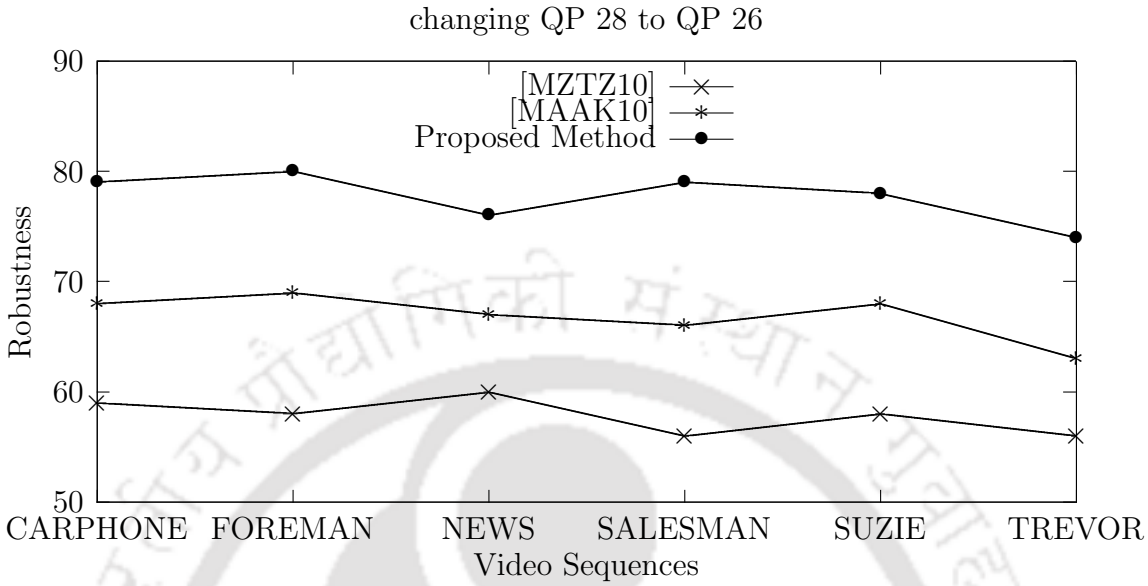


Figure 5.20 Average robustness against changing QP 28 to QP 30 without using location map.

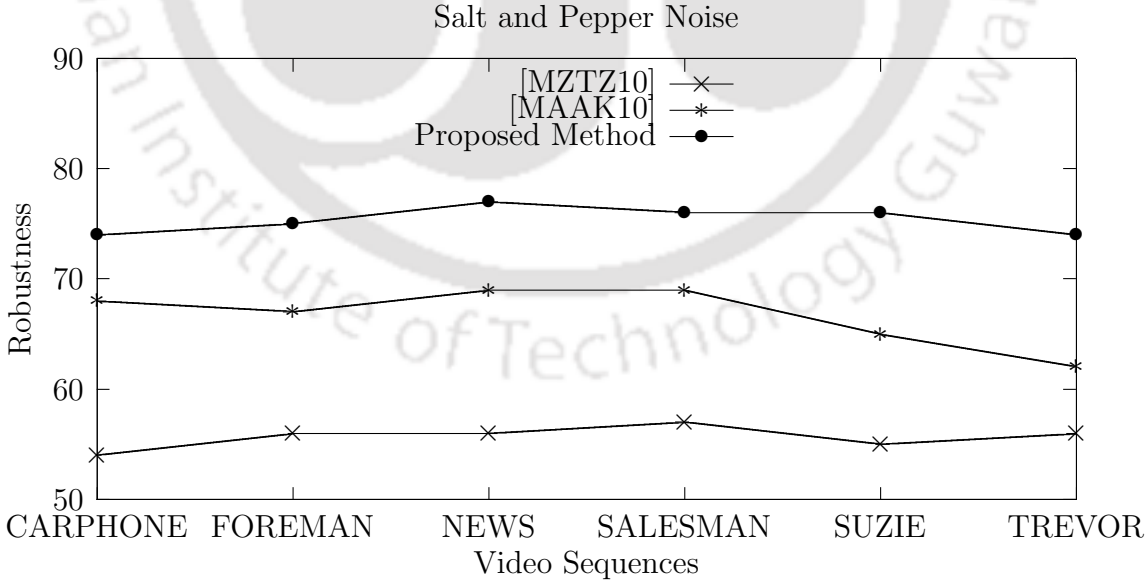


Figure 5.21 Average robustness against salt and pepper noise without using location map.

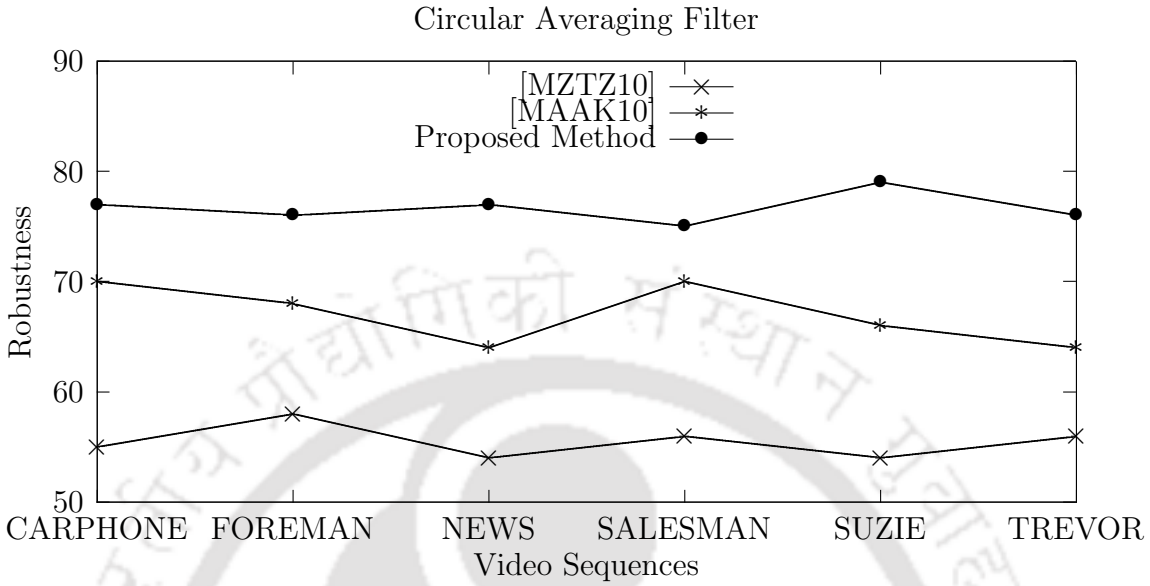


Figure 5.22 Average robustness against circular averaging filter without using location map.

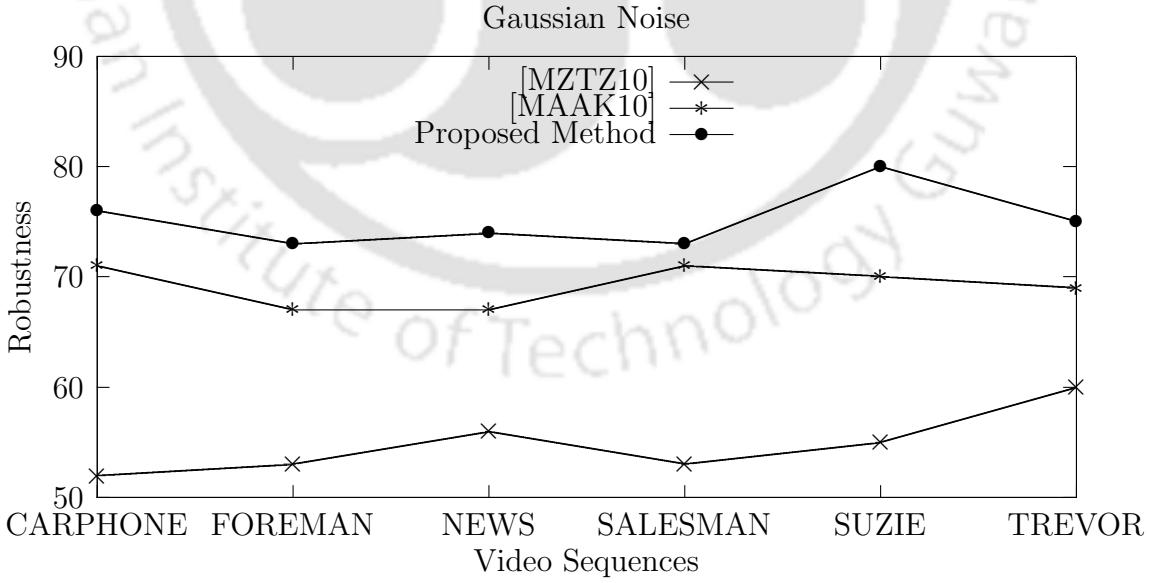


Figure 5.23 Average robustness against gaussian noise without using location map.

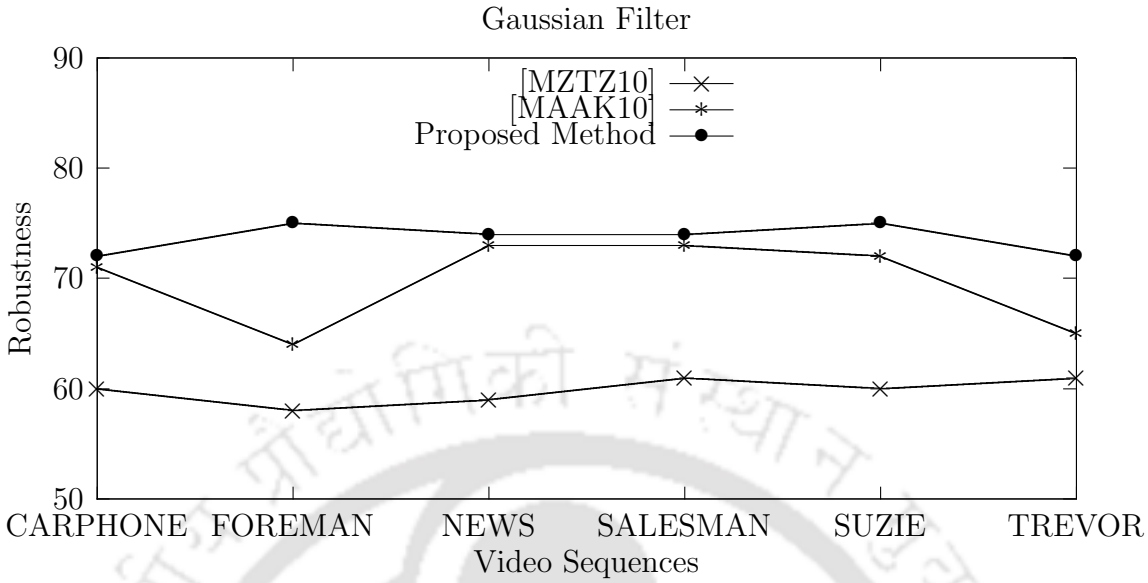


Figure 5.24 Average robustness against gaussian filter without using location map.

to solve this problem in a GOP, but there may be motion incoherent GOPs, which can be utilized to mount MC-TFA attack [PDB09]. In the next chapter, motion coherent watermarking in compressed domain to resist inter frame collusion attack is presented.

Chapter 6

CRMC: Collusion Resistant Motion Coherent Watermarking

Video watermarking technique has to be robust; subsequent processing of watermarked data should not impair the detection of embedded information [PB06]. Video signals are susceptible to image and video processing attacks and hostile attacks like collusion attack, copy attack, re-encoding attack, and frame drop or alter attack. In the previous chapter, it is observed that watermark estimation attacks (such as collusion attack, copy attacks, etc.) are a potential threat to the conventional video watermarking methods. The collusion attack, mentioned in [PDB09], is based on motion compensated temporal frame averaging (MC-TFA). This attack [PDB09] cannot be prevented when watermarking is performed in frame-by-frame manner. The prevention of distortion drift helps to solve this problem within a GOP, however there may be motion incoherent GOPs which can be prone to MC-TFA attack [PDB09].

A collusion attack is a type of watermark estimation attack. For preventing watermark estimation attacks, every homogeneous video regions preferably carry same watermark sample, whenever and wherever it appears in the video. In collusion attack, watermarked video frames are analyzed or combined with the goal of removing the

watermark. Two types of inter-frame collusion attacks are possible: different videos with same watermark, *i.e.*, Type I collusion attack and copies of the same video with different watermarks are combined to produce unwatermarked video, *i.e.*, Type II collusion attack. Inter-frame collusion attacks exploit the inherent redundancy within video frames or in watermark to produce the unwatermarked copy of the video [PB06].

It is important to protect video watermarks from motion compensated temporal frame averaging (MC-TFA) [PB06, DD04] along with conventional temporal frame averaging (TFA) [LH07, SKH05]. Successive frames in a short video neighborhood (SVN) generally have a high correlation so it is possible to align and average them to obtain a perceptually similar video stream. The uncorrelated watermark samples are averaged in the process. Moreover, if embedded watermark overlooks motion information then it may result in destruction of embedded watermark, which is critical. To prevent collusion attack, the same watermark sample would be carried by each point of video signal along the motion axis such that motion compensated temporal frame averaging no longer has any impact.

Interestingly, in some early works on video watermarking, Hartung and Girod have ensured that a watermark is motion coherence within a group of pictures (GOP) with the objective to cancel drift compensation [HG98]. Use of this simple strategy does not completely ensure full motion coherency in the video. However, if watermark embedded in different I-frames are not motion coherent among themselves then the watermark will lose its motion coherency from one GOP to another.

Budhia *et al.* have applied a block based motion compensation before temporal frame averaging which highlight that the signal of interest is basically the prediction error after motion compensation [UBZ06]. Khalilian and Bajic [KB11] have used the statistics of the first principle component of wavelet transform applied on each plane of the pixel cubes to make the watermarking system robust to collusion attacks. However, the given experimental results in [UBZ06, KB11] did not clearly exhibit a significant

improvement.

In [LH07], authors have designed a content dependent watermark to deal with watermark estimation attacks in compressed domain. Noorkami [NM07] have proposed a self-collusion resistant method using a combination of public and private key. Vinod *et. al.* [PDB09] have designed a system to check whether a video sequence contains any motion incoherent component using features extracted from frame prediction error.

In compressed video streams, it is difficult to predict the impact of change in motion information. Moreover, if the watermark is inserted into the compressed video stream then embedded watermark is exported in all frames of that GOP during the decoding process via motion compensation. In compressed domain, therefore resistance from collusion attack is a tedious job. In this connection, the algorithms proposed in [LH07, NM07] are not robust against motion compensated temporal frame averaging (MC-TFA) attack. To the best of our knowledge, no motion coherent watermarking method existing in literature, which is designed for compressed domain to prevent the collusion attack.

The rest of the chapter is organized as follows. In the next section, the motivation of this work is described. In Section 6.2, a watermarking method is proposed that can resist the aforesaid collusion attack. Detection of motion coherent blocks and embedding and extraction of watermark bits in those blocks are described. The simulation results are shown in Section 6.3. The chapter is concluded in Section 6.4.

6.1 Motivation

Motivated by the aforesaid limitations of the existing literature, a novel method has been proposed in this chapter to detect motion coherent similar blocks in successive frames in a short video neighborhood using compressed domain features, like, luminance intra prediction modes and chrominance prediction modes with respect to motion coherency as well as static similarity. Similar watermark is embedded in homogeneous

regions. A collusion resistant watermarking method is proposed and robustness against collusion attack and other image processing attacks are evaluated. The luminance intra prediction modes and chrominance prediction modes are denoted by luma modes and chroma modes, respectively in the rest of this chapter.

6.2 Proposed Method

The goal of this work is to design an algorithm that can prevent the collision attack [PDB09] for H.264 compressed video. The collusion attack proposed in [PDB09] is based on motion compensated temporal-frame averaging (MC-TFA), which is described in Section 2.4.5. The block diagram of the proposed embedding and extraction methods are depicted in Figure 6.1 and Figure 6.2, respectively.

Within a *short video neighborhood* (SVN), most video frames are visually coherent when no scene change is detected. A *similar region* is defined as a subpart of a frame containing blocks that are motion coherent within a SVN. A method for detection of such similar regions in compressed domain is the one of the main contributions of this work. Since the proposed method is in compressed domain, decoding and re-encoding of the video is not required and thus the method is computationally less expensive. Motion coherent blocks in the similar region are further sub-divided into two sub-regions, *i.e.*, *motion similar region* and *static similar region*. A motion coherent block having motion vector greater than a prescribed threshold is regarded as the motion similar region. Similarly, a motion coherent block with no motion or motion vector less than the threshold is regarded as the static similar region. Blocks in similar region are clustered based on luma modes and chroma modes of I-frames within a SVN. Static similar region and motion similar region are clustered separately.

Video frames are partitioned into non-overlapping blocks of size 4×4 within a SVN. A block denotes a 4×4 block in the rest of this work. Motion vector, motion threshold, luma mode, and chroma mode of a block $B(f, i, j)$ in the current frame f are denoted

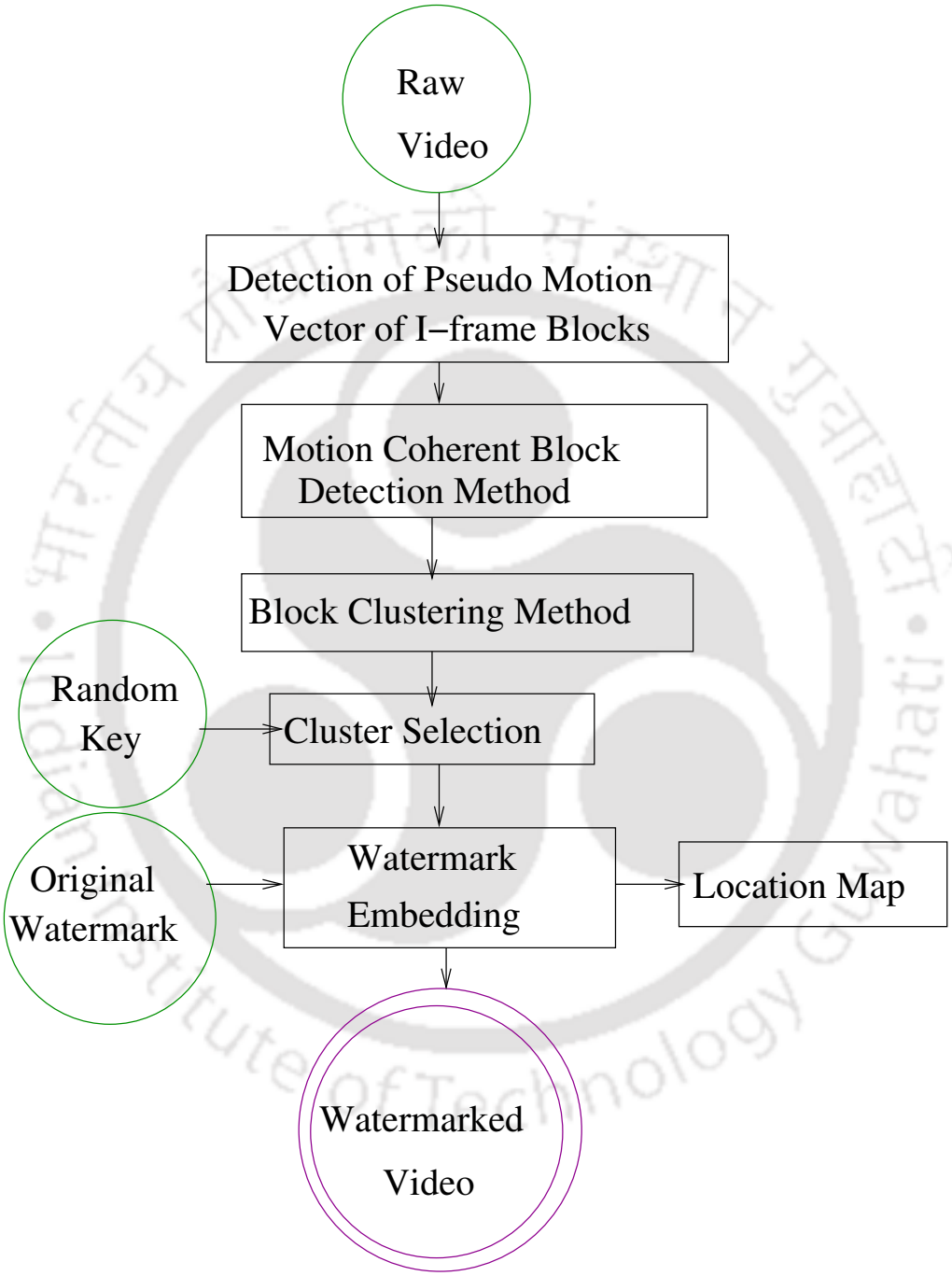


Figure 6.1 Block diagram of the proposed embedding method.

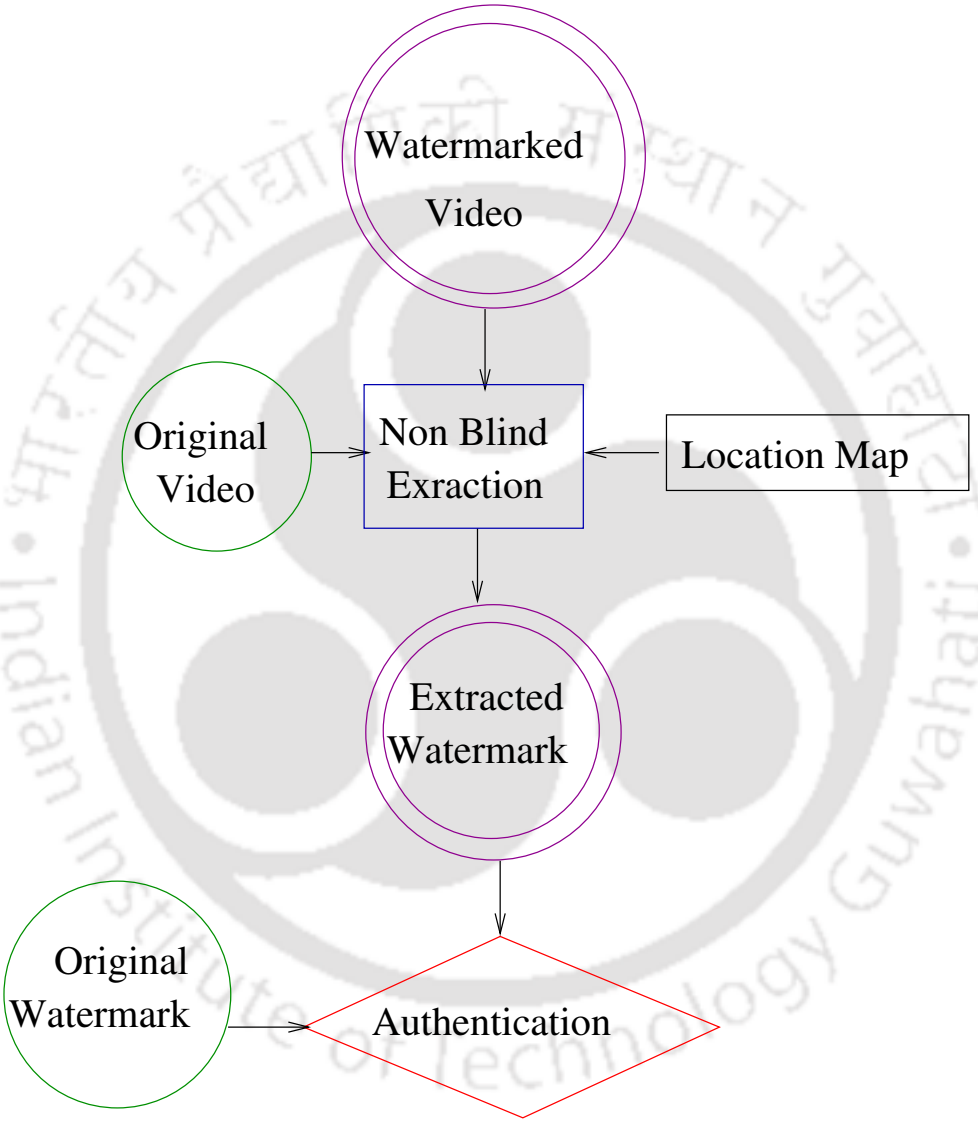


Figure 6.2 Block diagram of the proposed extraction method.

by $MV(f,i,j)$, MV_{th}, $P(f,i,j)$, and $M(f,i,j)$, respectively. The current I-frame and its consecutive previous I-frame are denoted by f and $f-k$, respectively, where k is the length of the GOP. H.264/AVC usually has only one I-frame in each GOP. During encoding of the I-frame, first color transform is performed to change from RGB space (red, green, and blue) to YCbCr space, where Y is the luminance component and Cb and Cr are the chrominance components.

In this work, there are two main modules. First, motion coherent blocks are clustered and next a suitable watermarking method is adopted for such coherent blocks to resist MC-TFA based collusion attack. In first three subsections, the process of clustering motion coherent blocks is performed. In Section 6.2.1, motion coherent blocks are detected after estimation of pseudo motion vector for I-frames. The detection of motion coherent blocks in similar region is presented in Section 6.2.2. The blocks resided in the similar region are clustered in Section 6.2.3. In Section 6.2.4, the watermark embedding and extraction methods are discussed. The similarity measure between the original watermark and the extracted watermark is given in Section 6.2.5.

6.2.1 Detection of Pseudo Motion Vector of I-frame Blocks

There exist no motion information (*e.g.* motion vector) in I-frame as it is intra coded. In the literature, different authors have suggested different ways to find pseudo motion vector for I-frames. For reliable detection of the motion of a block in an I-frame, a pseudo motion vector of that block is derived from the past reference P-frames within a SVN using the compressed domain parameters. The complete process is narrated using five steps as follows:

Step 1: P-frames are divided into non-overlapping blocks. Motion vectors for all blocks in P-frames are calculated. Normalization of each motion vectors is performed to ensure that it point directly to the location in the immediate previous frame [LLZ07]. It simplifies the referencing relationship since H.264/AVC supports multiple reference

frames. All normalized motion vectors are smoothen by using a 3×3 median filter [DXZF11].

Step 2: Since the value of motion vector of intra-coded blocks are zero in P-frames, motion vector for intra-coded blocks are estimated from neighboring blocks and finally forms a complete motion vector field is formed [WS03].

Step 3: Motion vector $MV(f, i, j)$ is assigned to each block in I-frames by interpolating motion vectors at the same location in the nearest P-frames.

Step 4: Accumulation operator [LLZ07] is applied to all motion vectors to enhance coherent motion and suppress noisy motion. It may increase non-zero motion vectors so remove those accumulated motion vectors that have a magnitude of zero before accumulation.

Step 5: Discontinuity in motion magnitude and direction is checked using spatial and temporal confidence measure [WZZ00].

Finally the pseudo motion vector $MV(f, i, j)$ for each block in I-frames are obtained.

6.2.2 Motion Coherent Block Detection Method

After the estimation of pseudo motion vectors $MV(f, i, j)$ for I-frames, motion coherent blocks are tracked in compressed domain within a SVN. The method for detection of motion coherent blocks in a similar region (motion similar region or static similar region) are proposed in Algorithm 7. A block $B(f, i, j)$ in current I-frame f in motion similar region has its motion coherent block $B(f-k, i', j')$ in previous I-frame $f-k$. Similarly, a block $B(f, i, j)$ in current I-frame f in the static similar region has its motion coherent block $B(f-k, i, j)$ in previous I-frame $f-k$. A block having motion vector value less than MV_{th} is regarded as negligible (blocks with a noisy motion vector) and the corresponding blocks are included in the static similar region. The rest of the blocks will be checked for motion similar region.

The red color edge box in Figure 6.3(a) shows a motion coherent block in motion

Algorithm 7: Motion Coherent Region Detection

Input: Blocks in I-frames within a SVN

Output: Similar regions (in motion and static)

while all blocks in I-frames are not exhausted within a SVN **do**

Calculate pseudo motion vector $MV(f,i,j)$ and motion coherent blocks (Section 6.2.1)

if $(|MV(f,i,j)| > MV_{th})$ **then**

/* Detection of blocks in motion similar region */

Calculate $P(f,i,j)$ and $M(f,i,j)$ of $B(f,i,j)$

Calculate $P(f-k,i',j')$ and $M(f-k,i',j')$ of $B(f-k,i',j')$

if $(P(f,i,j) = P(f-k,i',j')$ **and** $M(f,i,j) = M(f-k,i',j'))$ **then**

$B(f,i,j)$ and $B(f-k,i',j')$ are in motion similar region

end

else

/* Detection of blocks in static similar region */

Calculate $P(f,i,j)$ and $M(f,i,j)$ of $B(f,i,j)$

Calculate $P(f-k,i,j)$ and $M(f-k,i,j)$ of $B(f-k,i',j')$

if $(P(f,i,j) = P(f-k,i,j)$ **and** $M(f,i,j) = M(f-k,i,j))$ **then**

$B(f,i,j)$ and $B(f-k,i,j)$ are in static similar region

end

end

end

similar region in 57^{th} frame (I-frame of 9^{th} GOP) and Figure 6.3(b) indicates the same block in 64^{th} frame (I-frame of 10^{th} GOP) in the *hall monitor* video. The red color edge box of Figure 6.4(a) shows a motion coherent block in the static similar region in 1^{st} frame (I-frame of 1^{st} GOP) and Figure 6.4(b) indicates the same block in 92^{nd} frame (I-frame of 14^{th} GOP) in the *highway* video.

6.2.3 Block Clustering Method

After detection of motion coherent blocks in similar regions for all I-frames within a SVN, blocks are merged to form unique clusters. Blocks in motion similar region and static similar region are clustered separately. Merging of blocks in similar region is performed using the proposed block clustering method (Algorithm 8) in compressed domain. A sliding window of size 3×3 blocks is used for clustering. The window slides in horizontal and vertical direction within a frame. To assign the cluster number of a block, the block is placed at the center of the window of the similar region. The value of all coefficients and motion vector in a zero block is zero.

The proposed block clustering method is illustrated in Figure 6.5 where a sliding window of size 3×3 blocks is shown in the pink color box. The block whose cluster number needs to be decided placed at the center of the window in brown color box. The rest of the blocks in the window are 8-neighbor blocks of the center block. The boundary blocks of a frame are padded with blocks having zero valued coefficients to include original boundary blocks in the center of the window. Such zero blocks are shown in blue color. Assume, $P(z)$ and $M(z)$ denote the luma mode and the chroma mode of the block z. In Figure 6.5(a), the block A is a boundary block. It can be included in the center of the window after only after padding zero blocks. The window then slides to check the rest of the blocks in Figure 6.5(b). The windows containing blocks B and C in Figure 6.5(a) slide vertically and horizontally, respectively in Figure 6.5(b). Similarly, the window containing block D slides horizontally and vertically over the

frame as shown in Figure 6.5.

6.2.4 Embedding and Extraction

To prevent motion coherent temporal frame averaging based collusion attack [PDB09], similar watermark should be embedded in similar motion coherent clusters of I-frames. In the proposed method, a bipolar watermark sequence (0,1) is embedded in I-frames using the techniques proposed in [CKLS97], where the length of the invisible watermark sequence is same as the number of clusters generated. The watermark embedding is done in all nonzero quantized AC coefficients in all blocks in each cluster of I-frames. However, the clusters are selected using a random key (refer to Section 6.2.6). Zero valued coefficients are not perturbed to avoid increase in the video bit rate. The embedding of watermark bit (W_i) in an AC coefficient C is performed as follows:

$$C^W = C(1 + \alpha W_i), \qquad C \neq 0 \tag{6.1}$$

where α and C^W denote the watermark strength and watermarked coefficient. The range of α is between 0 to 1.

The watermark extraction is performed at the decoder after entropy decoding. The extraction procedure is exactly the reverse process of watermark embedding. The watermark is extracted as follows:

$$W_i^* = \frac{C^W - C}{\alpha}, \qquad C^W \neq 0, \;\; C \neq 0 \tag{6.2}$$

where W_i^*, C^W, and C denote extracted watermark, coefficient of the watermarked video, and original coefficient, respectively.

Algorithm 8: Block Clustering Method

Input: Similar regions (in motion and static)

Output: Clusters with cluster number

while (all I-frames are not exhausted within a SVN) **do**

All boundaries of a frame are padded with zero blocks to include boundary blocks at the center of the sliding window

while (all non- overlapping blocks are not exhausted) **do**

Estimate motion coherent blocks (static and motion) in similar region (Algorithm 7)

Slide the window over the frame

A block, say, $B1$ is in similar region

if ($B1$ is not included in any cluster) **then** Place $B1$ at the center of the window

Another block, say, $B2$ is also in that similar region and in 8-neighbor of $B1$

if ($P(B1) = P(B2)$ **and** $M(B1) = M(B2)$) **then** $B1$ and $B2$ in same cluster

if ($B2$ is already included in a cluster) **then** $B1$ is in that cluster **end**

if ($B1$ and $B2$ are not in any cluster) **then** $B1$ and $B2$ form new cluster **end**

end

end

end

For each cluster **do** Apply bounding box split method [DXZF11] for largest set of blocks over at least three I-frames (motion similar region) or all I-frames (static similar region)

All spatially disjoint clusters are marked with a distinct cluster number

end

end

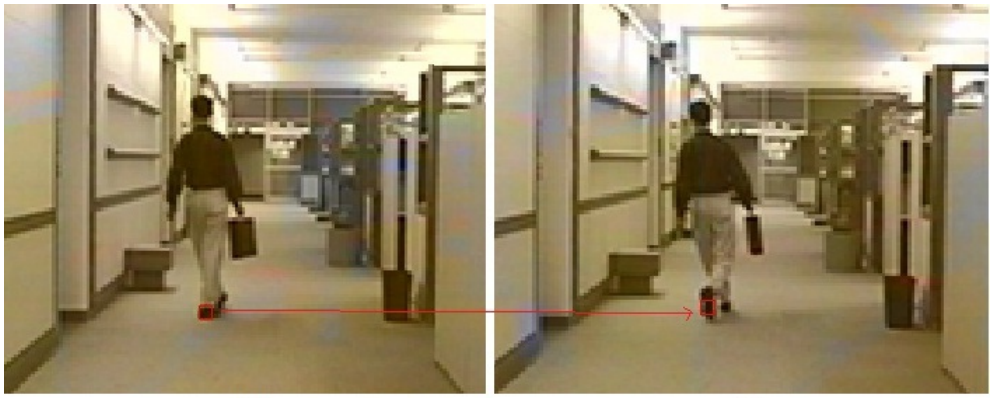


Figure 6.3 A motion coherent block in motion similar region in the hall monitor video.

Figure 6.4 A motion coherent block in static similar region in the highway video.

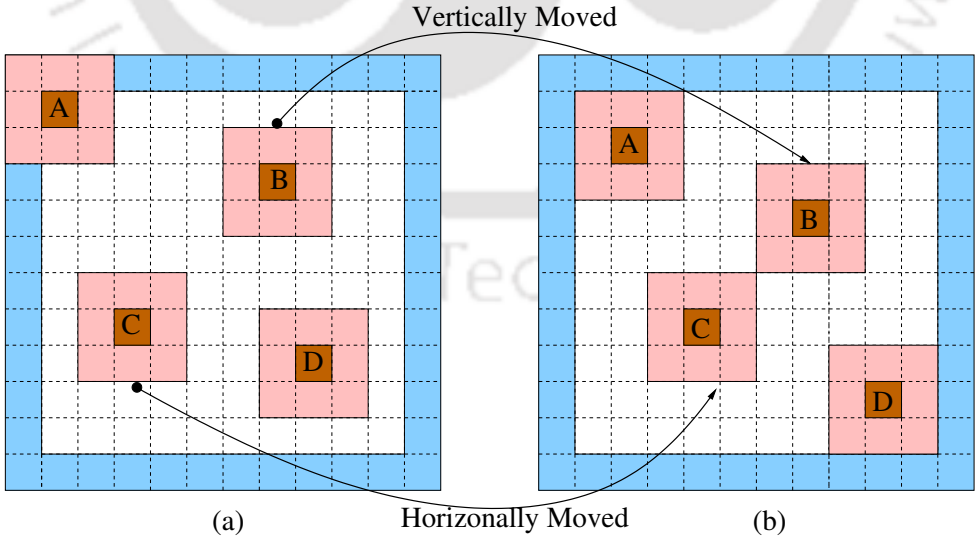


Figure 6.5 The diagonal (horizontal and vertical) movement of window containing blocks A and D, vertical movement of window containing block B, and horizontal movement of window containing block C to form clusters.

6.2.5 Watermark Similarity Measure

It is highly unlikely that the extracted watermark W^* will be identical to the original watermark W. Even the process of recompressing the watermarked video for delivery will cause W^* to deviate from W. The similarity of W^* and W is measured in [CKLS97] as

$$sim(W, W^*) = \frac{W^*.W}{\sqrt{W^*.W^*}} \tag{6.3}$$

where $sim(W, W^*)$ denotes the similarity measure between W and W^*. The distribution on $W^*.W$ is computed in [CKLS97] as follows:

$$\sum_{i=1}^{n} W_i^*.W_i \tag{6.4}$$

where W_i and W_i^* are the i^{th} watermark bit in the original watermark sequence W and extracted watermark sequence W^* and n is number of bits in the original watermark sequence W. Similarly, $W^*.W^*$ is computed as follow:

$$\sum_{i=1}^{n} W_i^{*2}. \tag{6.5}$$

If $sim(W, W^*) > S_{th}$, where S_{th} is the similarity threshold, then the extracted watermark sequence is similar to the original watermark sequence. In this work, it is assumed that if $S_{th} \geq 60\%$ then the original watermark W and the extracted watermark W^* are same. If $sim(W, W^*) < 60\%$, then the watermarked video is discarded. With the increase in similarity measure, the robustness of the proposed method increases.

6.2.6 Security

In the proposed method, the security of the watermarking algorithm lies in selecting clusters randomly using a key. Assume that the total number of clusters is G and the each cluster has a number of motion coherent blocks on an average. If the payload is

b, then the cryptographic space for selecting O clusters is given as follows:

$$b = a \times^{G} \mathrm{C}_{O} \tag{6.6}$$

If a cluster is selected, then all the blocks in that cluster are watermarked using the method described in Section 6.2.4.

6.3 Experimental Results

The proposed method is implemented using H.264/AVC [Ric10] reference software JM 17.2 [S08]. Table 6.1 gives experimental setup. The search range for motion estimation is [-32, 32]. Three reference frames are used, the fast full search in JM is employed, and all intra and inter prediction sizes are enabled.

Table 6.1 Experimental Setup

Parameters	**Values**
Video Format	Quarter Common Intermediate format (QCIF)
Frame Resolution	176×144
Frame Rate	30 frames per second
Codec Used	H.264/AVC reference software JM 17.2
Intra Period	7
GOP Structure	IBPBPBP
Encoding Profile	High Profile
Entropy Encoding	CAVLC
Number of frames encoded	140 frames per video
Quantization Parameter (QP)	28 (I-frames and P-frames)
Payload	50,100,150
Motion Threshold (MV_{th})	2
Video Sequence	Carphone, Foreman, Hall Monitor, Salesman, Suzie, Highway, Bridge Close, Bridge Far

In the experimentation, motion threshold (MV_{th}) is taken as 1. Motion coherent clusters in the static similar region and motion similar region are shown in the *foreman* video within a SVN in Figure 6.6 and Figure 6.7, respectively. Green color edge boxes in Figure 6.6(a) are blocks in static similar region after clustering in 8^{th} frame (I-frame

Figure 6.6 Blocks in static similar region after clustering in I-frame of 2^{nd} GOP and I-frame of 7^{th} GOP in the foremen video.

of 2^{nd} GOP). In Figure 6.6(b), the same blocks in 43^{th} frame (I-frame of 7^{th} GOP) in the *foremen* video are shown. Blue color edge boxes in Figure 6.7(a) are blocks in motion similar region after clustering in 1^{st} frame (I-frame of 1^{st} GOP). In Figure 6.7(b), the same in 8^{th} frame (I-frame of 2^{nd} GOP) in the *foremen* video are shown.

Figure 6.7 Blocks in motion similar region after clustering in I-frame of 1^{st} GOP and I-frame of 2^{nd} GOP in the foremen video.

Figure 6.6 indicates mostly smooth background areas which are common over all frames. For clusters in the static similar region, I-frames of two separated GOPs (2^{nd}

and 7^{th}) are shown. These clusters are of different sizes, both large clusters as well as small clusters containing single block. Figure 6.7 illustrates areas having motion. For motion coherent clusters in motion similar region, I-frames of two consecutive GOPs (1^{st} and 2^{nd}) are presented. These clusters are generally small in size containing mostly two or three blocks in each cluster. Size and number of clusters are different for different videos. It is clear from the results that quite sufficient number of block clusters are available for embedding.

Video Quality Metric, Peak Signal-to-Noise Ratio, and Bit Increase Rate of the proposed method at payload={50,100,150} [refer to Appendix] for an average of 140 frames for video sequences are depicted in Table 6.2. Video Quality Metric in Table 6.2 is in the order of 10^{-2}. It indicates that watermarked videos have acceptable visual quality. Moreover, Peak Signal-to-Noise Ratio also confirms that the proposed embedding method does not sacrifice the visual quality very much. In case of Bit Increase Rate, the increase in video rate is in the order of 10^{-3} which is nominal.

Figure 6.8(a) illustrates the average number of motion coherent blocks for each video sequence. The average robustness [refer to Appendix] against inter-frame collusion attack or MC-TFA and circular averaging filter providing location map to the decoder for extraction, where collusion window [3×3], $\tau_{out} = 100$ [PDB09], and Circular averaging filter r = 0.05, are shown in Figure 6.8(b) and Figure 6.8(c), respectively. Figure 6.9 depicts the average robustness against salt and pepper noise, gaussian filter, and additive white gaussian noise providing location map to the decoder for the proposed method, where Gaussian Noise density = 0.001, Salt and Pepper Noise density = 0.001, Gaussian Filter [5×5], and sigma=0.3 [MAAK10]. The robustness results against MC-TFA attack are acceptable. However, the results can be improved by further minimizing the synchronization error.

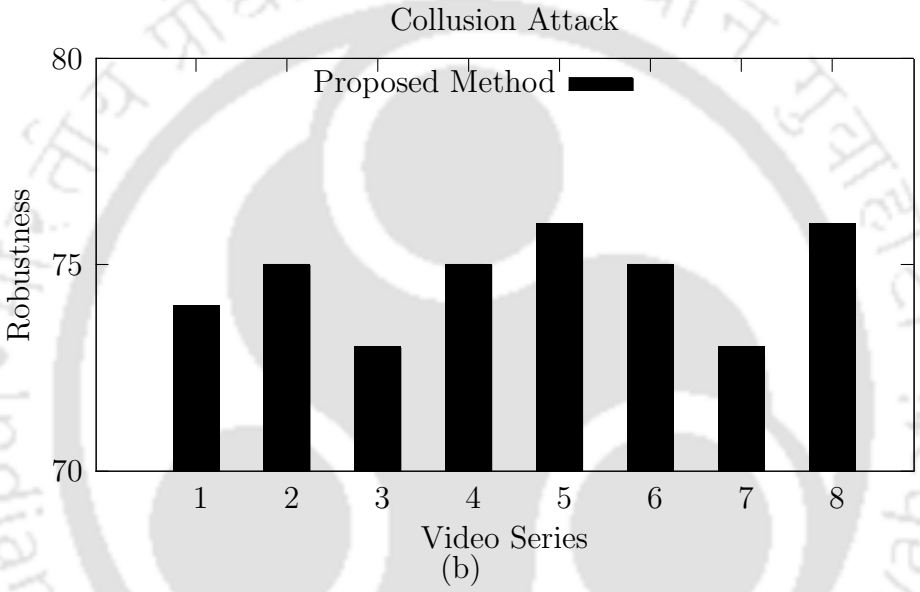
The proposed work is not compared with any related work, because no literature is available for searching motion coherent regions in compressed domain as mentioned

Table 6.2 Results for VQM, PSNR, and BIR of the proposed method.

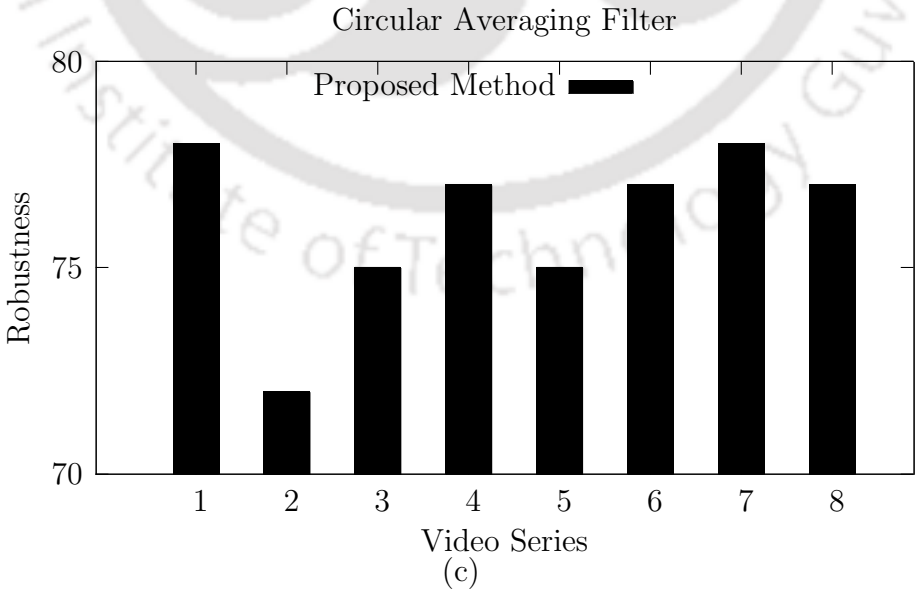
Sequence	Payload	VQM$\times 10^{-2}$	PSNR (dB)	BIR$\times 10^{-3}$
Carphone	50	2.72	37.11	1.66
	100	2.96	37.06	0.65
	150	3.11	36.98	1.2
Foreman	50	2.73	35.78	2.2
	100	4.12	35.43	0.49
	150	4.86	35.04	0.57
Hall Monitor	50	2.69	36.6	0.97
	100	2.75	36.51	-0.08
	150	3.02	36.23	-4
Bridge Close	50	3.09	36.01	1.3
	100	3.21	35.94	1.0
	150	3.40	35.86	1.9
Salesman	50	4.60	36.04	1.8
	100	4.66	35.95	1.1
	150	4.02	35.69	1.6
Suzie	50	3.43	36.41	1.3
	100	3.62	36.01	2.0
	150	3.85	35.83	2.1
Highway	50	2.83	36.21	1.6
	100	3.12	35.91	1.2
	150	3.84	35.33	2.2
Bridge Far	50	3.51	35.61	1.7
	100	3.71	35.41	-2.1
	150	4.06	35.03	2.4

Motion Coherent Blocks		
Video Series	Video Sequence	Number of Blocks
1	Carphone	161
2	Foreman	236
3	Hall Monitor	189
4	Bridge Close	175
5	Salesman	191
6	Suzie	209
7	Highway	110
8	Bridge Far	127

(a)



(b)



(c)

Figure 6.8 Part (a) of the figure shows average number of motion coherent blocks for each video sequence. Part (b) and part (c) of the figure depict the average robustness against collusion attack and circular averaging attack.

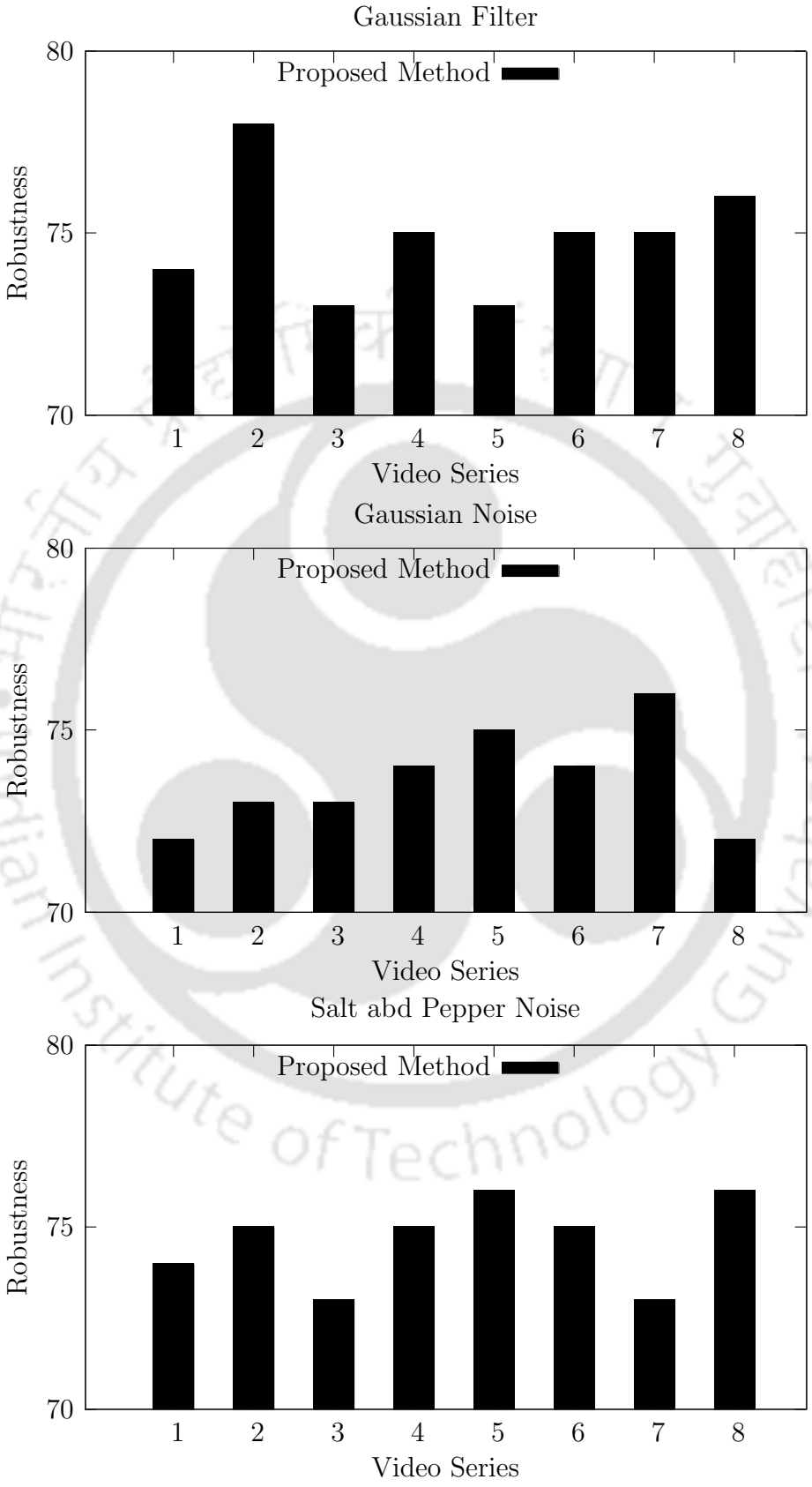


Figure 6.9 Average robustness against gaussian filter, gaussian noise, and salt and pepper noise.

earlier. Moreover, for a fair comparison, the proposed method is not compared with any spatial domain methods due to their high computational complexity.

6.4 Summary

In this work, a compressed domain motion coherent cluster detection method for H.264/AVC is proposed. The motion coherent blocks are detected based on the pseudo motion vectors of I-frames. Such blocks are grouped together based on luma modes and chroma modes. Motion coherent regions are merged into distinct clusters using the proposed clustering method. A watermarking method using these motion coherent regions is presented, where blocks in similar clusters are embedded with same watermark bits and blocks in different clusters are watermarked with different embedding bits along the temporal axis. This makes the watermarking method robust to inter-frame collusion attack. A block for watermark embedding is detected based on motion coherency so that the frame averaging will not be able to remove the watermark bit. Thus a robust watermarking method is designed based the proposed motion coherent similar region detection method. An exhaustive experimentation is performed to justify the efficiency of the proposed method. The result of this work appears in [DSN13b, Dut13].

Chapter 7

Conclusion and Future Directions

In the video watermarking research paradigm, one of the major concerns is robustness against intentional and unintentional watermarking attacks such as watermark estimation attacks, re-synchronization and compression errors, scaling, etc. In this dissertation, the primary motivation is to enhance the robustness of the existing video watermarking methods against such attacks. Problems are formulated by taking into account of few interesting observations of existing P-frame as well as I-frame based watermarking methods. In this dissertation, different issues in compressed domain watermarking for H.264/AVC videos are addressed. Algorithms are designed to resolve some of these issues in the dissertation. A chapter-wise summary of the contributions made to solve the problems is presented below:

Contributions of Chapter 1

The Chapter 1 introduces the compressed domain video watermarking with the basics of H.264/AVC. The state of the art literature is briefly presented and finally the research motivation and problem statement of the dissertation is discussed.

Contributions of Chapter 2

In the first part of the PhD dissertation, a P-frames based watermark embedding algorithm with blind extraction process is proposed. The proposed method has controlled the increase in the video bit rate. The embedding algorithm is made robust to minimize synchronization error by appropriate selection of macroblocks based on spatial and temporal characteristics of partially decoded video parameters. In P-frames, luminance intra and inter prediction modes coexist. Based on luminance inter prediction modes, different block sizes are determined. There are 10 different prediction modes, *i.e.* $\{4 \times 4, 4 \times 8, 8 \times 4, 8 \times 8, 8 \times 16, 16 \times 8, 16 \times 16$, SKIP$\}$ exists in P-frames. It is observed in the literature that selecting macroblocks based only on modes in P-frame may not be appropriate as prediction modes are changed frequently due to re-encoding. In the proposed work, watermark bits are embedded by modifying nonzero quantized coefficients of the residual block to restrict increase in video bit rate. To prevent FDAS attack repeat accumulate code with an erasure channel is used for designing a robust watermark. The proposed algorithm is shown to be robust against image processing attacks. A comprehensive set of experiments is carried out to justify the above claims.

Contributions of Chapter 3

A crucial problem of decoder based compressed domain watermarking algorithms is error drift, which refers to the watermark error accumulations among different blocks during intra or inter predictions. In this part of the dissertation, the drift error propagation is resisted using a robust reversible watermarking algorithm technique. It is observed for P-frame encoding, different intra and inter prediction modes coexist and motion estimation is performed over five (maximum) reference frames. A small modification in a macroblock may propagate drift distortion error to a number of macroblocks in that frame or frames predicted from the modified macroblock, which may degrading

the visual quality of the watermarked video. Reversible watermarking allows to get back the original (unwatermarked) video coefficients at the decoder. The realization of different thresholds is performed to maintain higher perceptual quality and robustness. To prevent FDAS attack, reed solomon code is used.

Contributions of Chapter 4

It is observed in the literature that the existing I-frame based watermarking methods have some limitations such as perceptual distortion in the watermarked video, distortion drift, non-blindness, fragility, etc. In this work, some existing I-frame based works are accumulated in such a way that those pitfalls can be minimized and necessary modifications are incorporated to improve the overall performance of the proposed method over the existing methods. Public keys are extracted using robust compressed domain features to minimize the location map, which decreases the overhead of secure transmission of large location map. The self collusion attack is also resisted using extracted keys. Appropriate selection of blocks decreases synchronization error in I-frames. Watermark is embedded in nonzero coefficients to restrict the increase in video bit rate. Location aware and unaware extraction are designed.

Contributions of Chapter 5

In the final part of the dissertation, a novel motion coherent watermarking algorithm is proposed that can resist MC-TFA based collusion attack. Similar watermark must be embedded in similar region based on motion coherency to resist MC-TFA based collusion attack. First, the motion coherent blocks are detected based on the pseudo motion vectors of I-frames. Such blocks are grouped together based on luminance prediction modes and chrominance modes. Motion coherent regions are merged into distinct clusters using the proposed clustering method. Then, blocks in similar clusters

are embedded with same watermarks and blocks in different clusters are watermarked with different watermarks. This makes the watermarking method robust to inter frame collusion attack. Blocks are detected based on motion coherency so frame averaging will not be able to remove the embedding bits.

Limitation of the work and Future Scope

There are some possibilities to further extending the current work presented in this dissertation. Some of those are listed below:

- Testing the robustness of the proposed algorithm to severe attacks, such as, changing the video format, resolution, and aspect ratio.
- Exploring the compressed domain feature to determine the exact motion removing the camera motion, zooming, panning, tilting etc.
- Looking into security measures for watermarking as similar to what exists in cryptography so that the algorithms can be used in domains such as legal and military applications where high security is essential.
- Developing a watermarking algorithm that completely compensates the error propagation in frames when the watermark is embedded in the H.264 bitstream without using reversible watermarking techniques.
- Design of watermarking algorithms in such a way that the same algorithm may be used for both scalable and non scalable videos, e.g., H.264/SVC and H.264/AVC, respectively.

An Application Scenario

A brief description of an application scenario of the work in the thesis is elaborated in this section as follows:

Streaming media: Streaming media broadcasts can also take the advantage of watermarking. One way to do it is to simply encode it with the source video signal, the same way a TV broadcaster would. Many encoding tools and some video capture cards now include the ability to burn a watermark directly and indelibly into an encoded video stream. But such burned-in watermark cannot be replaced in case if re-branding of syndicated content is required. Keeping the video file intact and overlay the watermark non-destructively at delivery time is possible by dynamic assembly of compressed media elements on the fly. In such a scenario, the use of robust and reversible watermarking technique for P-frames can be helpful. P-frame watermarking can provide better perceptual quality of the watermarked video. Since the algorithm is robust against attacks, the media will remain secured and can be authenticated. In addition, the reversible nature will help in re-branding of syndicated content if required. Furthermore, compressed domain watermarking will help to reduce computational complexities and time delays, especially in case of live streaming.

Appendix

1 Payload

Payload is the actual number of watermark bits that are embedded in a video sequence.
– *Average payload* is the actual number of watermark bits that are embedded per frame in a video sequence.

2 Peak Signal to Noise Ratio (PSNR)

Peak signal-to-noise ratio, often abbreviated as PSNR, is the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation [HG08]. As many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale. It is most easily defined via the mean squared error (MSE) which for two $M \times N$ monochrome images I and K where one of the images is considered a noisy approximation of the other is defined as

$$\text{MSE} = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \|I(i,j) - K(i,j)\|^2 .$$

The PSNR is defined as

$$\text{PSNR} = 10. \log_{10} \frac{(I_{max})^2}{\text{MSE}} dB,$$

i.e.,

$$\text{PSNR} = 20. \log_{10} \frac{I_{max}}{\sqrt{\text{MSE}}} dB,$$

where I_{max} is the maximum possible pixel value of the image I. For a 8 bit gray scale

image since maximum gray value is 255, PSNR is defined as

$$\text{PSNR} = 20.\log_{10}\frac{255}{\sqrt{\text{MSE}}}dB.$$

– *Average PSNR* is the average of PSNR over different payloads for a fixed number of frames in a video.

3 Video Quality Metric (VQM)

Video quality metric (VQM) is used to provide an objective measurement for perceived video quality [vqm96]. It measures the perceptual effects of video impairments including blurring, noisy motion, global noise, block distortion, and color distortion and combines them into a single metric. This metric is between zero and one; zero means not having any distortion while one shows maximum impairment. Original compressed and watermarked sequences are used as original and processed clips, respectively.

– *Average VQM* is the average VQM over different payloads for a fixed number of frames in a video.

4 Embedding Capacity

Embedding capacity is the maximum number of watermark bits that can be embedded in a video sequence.

– *Average embedding capacity* is the maximum number of watermark bits that can be embedded per frame in a video.

5 Bit Increase Rate (BIR)

Bit increase rate (BIR) is defined as the percentage of bit rate increase per embedded bit [MAAK10].

$$\text{BIR} = \frac{|BR_{WM} - BR_{ORG}|}{\text{Payload} \times BR_{ORG}} \times 100,$$

where BR_{WM} and BR_{ORG} are the number of bits in watermarked and original video sequences, respectively.

– *Average BIR* is the average of BIR over different payloads for a fixed number of frames in a video.

6 Bit Error Rate (BER) and Robustness

Bit error rate (BER) is defined as the frequency of bit errors when detecting a multi-bit watermark message [DSB10], *i.e.*,

$$\text{BER} = \frac{\text{number of error bits}}{\text{total number of bits sent}}.$$

Robustness of a watermarking method [DMLH06] is given by

$$\text{Robustness} = (1\text{-BER}) \times 100.$$

– *Average robustness* is the robustness averaged over different payloads for a video sequence with fixed number of frames of a video.

7 Error Correcting Code's Capability

An error-correcting code (ECC) is a system of adding redundant data, or parity data to a message or repetition code, such that it can be recovered by a receiver even when a number of errors were introduced either during the process of transmission or on storage [LC04]. Error correcting code's capability is the capability of a code being used to recover a number of errors introduced either during the process of transmission or on storage.

7 Erasure Channel

Erasure channel in information theory and telecommunications is defined as a communication channel model wherein errors are described as erasures [Mac03]. In this model,

a transmitter sends a bit (a zero or a one), and the receiver either receives the bit or it receives a message that the bit was not received ("erased"). This channel is used frequently in information theory because it is one of the simplest channels to analyze. A erasure channel with erasure probability p is a channel with binary input, ternary output, and probability of erasure p. That is, let X be the transmitted random variable with alphabet $\{0, 1\}$. Let Y be the received variable with alphabet $\{0, 1, e\}$, where e is the erasure symbol. Then, the erasure channel with capacity $1 - p$ is characterized by the conditional probabilities:

$$\begin{aligned} Pr(Y = 0|X = 0) &= 1 - p \\ Pr(Y = e|X = 0) &= p \\ Pr(Y = 1|X = 0) &= 0 \\ Pr(Y = 0|X = 1) &= 0 \\ Pr(Y = e|X = 1) &= p \\ Pr(Y = 1|X = 1) &= 1 - p. \end{aligned}$$

8 Repeat-Accumulate (RA) Codes

Repeat-Accumulate (RA) Codes [Joh09] are a low complexity class of error-correcting codes. They were devised so that their ensemble weight distributions are easy to derive. In an RA code, an information block of length N is repeated q times, scrambled by an interleaver of size qN, and then encoded by a rate 1 accumulator. It as a block code whose input block $\{z_1, \ldots, z_n\}$ and output block $\{x_1, \ldots, x_n\}$ are related by the following formula:

$$x_1 = z_1 \quad \text{and} \quad x_i = x_{i-1} + z_i \quad \text{for} \quad i > 1.$$

The encoding time for RA codes is linear and their rate is $1/q$. They are non systematic.

9 Reed-Solomon (RS) Code

Reed-Solomon (RS) Code is considered as a maximum distance separable code [WB94,

SJM+04]. Various researchers have suggested to use this code as an efficient solution for packet loss protection [SJM+04]. RS codes c are denoted by their length b and dimension d as (b, d) codes. Each RS code word can be related to a system of q linear equations in d variables to get a unique solution is given as

$$\begin{aligned} c &= (c_0, c_1, ..., c_{q-1}) \\ &= [E(0), E(\alpha), \cdots, E(\alpha^{q-1})], \end{aligned}$$

where

$$\begin{aligned} E(0) &= m_0 \\ E(\alpha) &= m_0 + m_1\alpha + \cdots + m_{d-1}\alpha^{d-1} \\ &\vdots \\ E(\alpha^{q-1}) &= m_0 + m_1\alpha^{q-1} + \cdots + m_{d-1}\alpha^{(d-1)(q-1)}. \end{aligned}$$

Assuming that the exact locations of the errors are not known, it may be possible to construct all distinct systems of d expressions from the set of expressions in $\{E(0), E(\alpha), \ldots, E(\alpha^{q-1})\}$. In $\binom{q}{d}$ such systems, $\binom{w+d-1}{d}$ of which will give incorrect information symbols. If the majority vote is taken among the solutions to all possible linear systems, the correct information bits will be received subject to the condition, $\binom{w+d-1}{k} < \binom{q-w}{d}$. A RS code of length q and dimension d can thus correct up to w errors, where w is as follows:

$$w = \left\lfloor \frac{q-d+1}{2} \right\rfloor.$$

Bibliography

[AS07] F. Ahmed and M. Siyal. A Robust and Secure Signature Scheme for Video Authentication. In *International Conference on Multimedia and Expo*, pages 2126–2129, 2007.

[AYCK04] P. Atrey, W. Yan, E. Chang, and M. Kankanhalli. A hierarchical signature scheme for robust video authentication using secret sharing. In *Multimedia Modelling Conference*, pages 330–337, 2004.

[BLD03] Y. Bodo, N. Laurent, and J. Dugelay. Watermarking video, hierarchical embedding in motion vectors. In *ICIP*, volume II-3, pages 739–742, 2003.

[BLD04] Yann Bodo, Nathalie Laurent, and Jean-Luc Dugelay. A comparative study of different modes of perturbation for video watermarking based onmotion vectors. In *EUSIPCO*, pages 1501–1504, 2004.

[CC07] D. Coltuc and J. Chassery. Very fast watermarking by reversible contrast mapping. *IEEE Signal Process. Letter*, 14(4):255–258, 2007.

[CKLS97] I. Cox, J. Kilian, F. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *Image Processing, IEEE Trans. on*, 6(12):1673–1687, 1997.

[DCOP99] Frederic Deguillaume, Gabriela Csurka, Joseph O'Ruanaidh, and Thierry Pun. Robust 3D DFT Video Watermarking. In *Proceedings of SPIE*, volume 3657, page 113124, 1999.

[DD03] G. Doerr and J. Dugelay. New intra-video collusion attack using mosaicing. In *International Conference on Multimedia & Expo*, pages 505–508, 2003.

[DD04] G. Doërr and J. Dugelay. Secure background watermarking based on video mosaicing. In *SPIE Security, Steganography, and Watermarking of Multimedia Contents VI*, volume 5306, pages 304–314, 2004.

[DMLH06] J. Dittmann, D. Mega, A. Lang, and J. Herrera. Theoretical Framework for a Practical Evaluation and Comparison of Audio Watermarking Schemes in the Triangle of Robustness, Transparency and Capacity. *Springer Trans. on Data Hiding and Multimedia Security I*, 4300:1–40, 2006.

[DSB10] G. Motta D. Salomon and D. Bryant. *An Engineer's guide to Automated Testing of High-speed Interfaces*. Artech House, 2010.

[DSN13a] T. Dutta, A. Sur, and S. Nandi. A robust compressed domain video watermarking in P-frames with controlled bit rate increase. In *NCC*, pages 1–5, 2013.

[DSN13b] T. Dutta, A. Sur, and S. Nandi. Motion Coherent Region Detection in H.264 Compressed Videos. In *IEEE International Conference on Multimedia Expo*, pages 1–5, 2013.

[DSS98] Jana Dittmann, Mark Stabenau, and Ralf Steinmetz. Robust MPEG Video Watermarking Technologies. In *Proceedings of the ACM International Conference on Multimedia*, pages 71–80, 1998.

[Dut13] T. Dutta. Motion Compensated Compressed Domain Watermarking. In *ACM Multimedia*, pages 1039–1042, 2013.

[DXZF11] P. Dong, Y. Xia, L. Zhuo, and D. Feng. Real-time moving object segmentation and tracking for H.264/AVC surveillance videos. In *International Conference on Image Processing*, pages 2309–2312, 2011.

[EA11] E. Esen and A.A. Alatan. Robust Video Data Hiding Using Forbidden Zone Data Hiding and Selective Embedding. *IEEE Trans. on Circuits and Systems for Video Technology*, 21(8):1130–1138, 2011.

[EK01] S. Emmanuel and M. Kankanhalli. Mask-based interactive watermarking protocol for video. volume 4518, pages 247–258, 2001.

[FF10] R. Facciol and R.A. Farrugia. Robust Video Transmission Using Reversible Watermarking Techniques. In *IEEE Symposium on Multimedia*, pages 161–166, 2010.

[FW11] G. Feng and G. Wu. Motion vector and mode selection based fragile video watermarking algorithm. In *Anti-Counterfeiting, Security and Identification*, pages 73–76, 2011.

[GBW01] K. Gopalan, D. Benincasa, and S. Wenndt. Data embedding in audio signals. In *Aerospace Conference*, volume 6, pages 2713–2720, 2001.

[GL08] X. Gong and H. Lu. Towards Fast and Robust Watermarking Scheme for H.264 Video. In *IEEE Symposium on Multimedia*, pages 649 –653, 2008.

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

[HG08] Q. Huynh and M. Ghanbari. Scope of validity of PSNR in image/video quality assessment. *Electronics Letters*, 44(13):800–801, 2008.

[HM00] M. Holliman and N. Memon. Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes. *IEEE Trans. on Image Processing*, 9(3):432–441, 2000.

[HM05] Oztan Harmanci and M. Mihcak. Motion picture watermarking via quantization of pseudo-random linear statistics. 5960:1142–1150, 2005.

[HZC11] W. Huo, Y. Zhu, and H. Chen. A Controllable Error-Drift Elimination Scheme for Watermarking Algorithm in H.264/AVC Stream. *Signal Processing Letters, IEEE*, 18(9):535–538, 2011.

[JhG00] M. Amado J. hernandez and F. Gonzalez. DCT Domain Watermarking Techniques for Still Images: detector Performance Analysis and a New Structure. *IEEE Trans. on Image Processing*, 9:55–68, 2000.

[Joh09] S. Johnson. *Iterative Error Correction: Turbo, Low-Density Parity-Check and Repeat-Accumulate Codes*. Cambridge University Press, 2009.

[KB11] H. Khalilian and I. Bajic. Multiplicative video watermarking with semi-blind maximum likelihood decoding for copyright protection. In *PacRim*, pages 125–130, 2011.

[KDHM99] Ton Kalker, Geert Depovere, Jaap Haitsma, and Maurice J. Maes. Video watermarking system for broadcast monitoring. volume 3657, pages 103–112, 1999.

[KL10] T. Kuo and Y. Lo. A hybrid scheme of robust and fragile watermarking for H.264/AVC video. In *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, pages 1–6, 2010.

[KLL08] T. Kuo, Y. Lo, and C. Lin. Fragile Video Watermarking Technique by Motion Field Embedding with Rate-Distortion Minimization. In *IIHMSP*, pages 853–856, 2008.

[KP03] D. Kirovski and F. Petitcolas. Blind pattern matching attack on watermarking systems. *IEEE Trans. on Signal Processing*, 51(4):1045–1053, 2003.

[KHMV05] Mehmet Kucukgoz, ztan Harmanc, M. Kvan Mhak, and Ramarathnam Venkatesan. Robust Video Watermarking via Optimization Algorithm for Quantization of Pseudo-Random Semi-Global Statistics. 2005.

[LC04] S. Lin and D. Costello. *Second Edition Error Control Coding*. Prentice-Hall, Inc., 2004.

[LCLF02] Chun-Shien Lu, Jan-Ru Chen, H.-Y.M. Liao, and Kuo-Chih Fan. Real-time MPEG2 video watermarking in the VLC domain. In *ICPR*, volume 2, pages 552–555, 2002.

[LH07] C. Lu and C. Hsu. Near-Optimal Watermark Estimation and Its Countermeasure: Antidisclosure Watermark for Multiple Watermark Embedding. *IEEE Trans. on Circuits and Systems for Video Technology*, 17(4):454–467, 2007.

[LHC06] T. Lu, W. Hsu, and P. Chang. Blind Video Watermarking for H.264. In *Canadian Conference on Electrical and Computer Engineering*, pages 2353–2356, 2006.

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

[LLB98] Gerrit C. Langelaar, Reginald L. Lagendijk, and Jan Biemond. Real-Time Labeling of MPEG-2 Compressed Video. *Journal of Visual Communication and Image Representation*, 9(4):256–270, 1998.

[LLZ07] Z. Liu, Y. Lu, and Z. Zhang. Real-time spatiotemporal segmentation of video objects in the H.264 compressed domain. *Journal of Visual Comunication and Image Representation*, 18(3):275–290, 2007.

[LYK07] S. Lee, C. Yoo, and T. Kalker. Reversible Image Watermarking Based on Integer-to-Integer Wavelet Transform. *IEEE Trans. on Information Forensics and Security*, 2(3):321–330, 2007.

[MAAK10] A. Mansouri, A.M. Aznaveh, F.T. Azar, and F. Kurugollu. A Low Complexity Video Watermarking in H.264 Compressed Domain. *IEEE Trans. on Information Forensics and Security*, 5(4):649–657, 2010.

[Mac03] D. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[MC96] Jianhao Meng and Shih-Fu Chang. Tools for compressed-domain video indexing and editing. volume 2670, pages 180–191, 1996.

[MC02] B.G. Mobasseri and D. Cinalli. Watermarking of compressed multimedia using error-resilient VLCs. In *IEEE Workshop on Multimedia Signal Processing*, pages 320–323, 2002.

[MC04] Bijan G. Mobasseri and Domenick Cinalli. Reversible Watermarking using Two-way Decodable Codes. In *In Security, Steganography and Watermarking of MultimediaContents VI, Proceedings of SPIE*, volume 5306, page 397404, 2004.

[MKHD99] M. Maes, T. Kalker, J. Haitsma, and G. Depovere. Exploiting shift invariance to obtain a high payload in digital image watermarking. In *IEEE International Conference on Multimedia Computing and Systems*, volume 1, pages 7–12, 1999.

[MSB02] Xiamu Niu Martin, Martin Schmucker, and Christoph Busch. Video Watermarking Resisting to Rotation, Scaling, and Translation. In *Proc. SPIE Security Watermarking of Multimedia Contents IV*, volume 5681, pages 512–519, 2002.

[MWS06] D. Marpe, T. Wiegand, and G. Sullivan. The H.264/MPEG4 Advanced Video Coding Standard and its Applications. *Communications Magazine*, 44(8):134–143, 2006.

[MZTZ10] X. Ma, Z.Li, H. Tu, and B. Zhang. A Data Hiding Algorithm for H.264/AVC Video Streams Without Intra-Frame Distortion Drift. *IEEE Trans. on Circuits and Systems for Video Technology*, 20(10):1320–1330, 2010.

[NM05] M. Noorkami and R. Mersereau. Compressed-domain video watermarking for H.264. In *International Conference on Image Processing*, volume 2, pages 890–893, 2005.

[NM07] M. Noorkami and R. M. Mersereau. A Framework for Robust Watermarking of H.264-Encoded Video With Controllable Detection Performance. *IEEE Trans. on Information Forensics and Security*, 2(1):14–23, 2007.

[NM08] M. Noorkami and R.M. Mersereau. Digital Video Watermarking in P-Frames With Controlled Video Bit-Rate Increase. *IEEE Trans. on Information Forensics and Security*, 3(3):441–455, 2008.

[NSAS06] Z. Ni, Y. Shi, N. Ansari, and W. Su. Reversible data hiding. *IEEE Trans. of Circuits System Video Technology*, 16(3):354–362, 2006.

[NTD06] C. Nguyen, Da. B. H. Tay, and G. Deng. A Fast Watermarking System for H.264/AVC Video. In *ASIA PACIFIC CONFERENCE ON CIRCUITS AND SYSTEMS*, pages 81–84, 2006.

[PB06] V. Pankajakshan and P. Bora. Motion-compensated inter-frame collusion attack on video watermarking and a countermeasure. *IEE Information Security*, 153(2):61–73, 2006.

[PDB09] V. Pankajakshan, G. Doërr, and P. Bora. Detection of motion-incoherent components in video streams. *IEEE Trans. on Information Forensics and Security*, 4(1):49–58, 2009.

[PRSM05] D. Profrock, H. Richter, M. Schlauweg, and E. Muller. H.264/AVC video authentication using skipped macroblocks for an erasable watermark. In *SPIE VCIP*, volume 5960, pages 1480–1489, 2005.

[QMH+04] G. Qiu, P. Marziliano, A. Ho, D. He, and Q. Sun. A Hybrid Watermarking Scheme for H.264/AVC Video. In *ICPR*, pages 865–869, 2004.

[Ric10] I. E. Richardson. *The H.264 Advanced Video Compression Standard*. Wiley Publication, 2010.

[SGD12] J. Stankowski, T. Grajek, and M. Domanski. Fast watermarking of MPEG-4 AVC/H.264 encoded HDTV video bitstreams. In *Picture Coding Symposium*, pages 265–268, 2012.

[SJM+04] K. Solanki, N. Jacobsen, U. Madhow, B.S. Manjunath, and S. Chandrasekaran. Robust image-adaptive data hiding using erasure and error correction. *IEEE Trans. on Image Processing*, 13(12):1627–1639, 2004.

[SKH05] K. Su, D. Kundur, and D. Hatzinakos. Statistical invisibility for collusion-resistant digital video watermarking. *IEEE Trans. of Multimedia*, 7(1):43–51, 2005.

[SLH04] Jiande Sun, Ju Liu, and Huibo Hu. Data hiding in independent components of video. In *Independent Component Analysis and Blind Signal Separation*, volume 3195, pages 970–976, 2004.

[SLRP11] K. Swaraja, Y.M. Latha, V.S.K. Reddy, and A.V. Paramkusam. Video watermarking based on motion vectors of H.264. In *INDICON*, pages 1–4, 2011.

[SP96] A. Said and W. Pearlman. An Image Multiresolution Representation for Lossless and Lossy Compression. *IEEE Trans. on Image Processing*, 5(9):1303–1310, 1996.

[SWC+11] P. Su, C. Wu, I. Chen, C. Wu, and Y. Wu. A practical design of digital video watermarking in H.264/AVC for content authentication. *Signal Processing: Image Communication*, 26(8-9):413–426, 2011.

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

[S08] K. Shring. H.264 Reference Software Group, 2008.

[TDSV+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 *ICIP*, volume 1, pages 407–410, 2000.

[THY09] P. Tsai, Y. Hu, and H. Yeh. Reversible image hiding scheme using predictive coding and histogram shifting. *Elsevier Signal Processing*, 89(6):1129–1143, 2009.

[Tia03] J. Tian. Reversible Data Embedding Using a Difference Expansion. *IEEE Trans. on Circuits System Video Technology*, 13(8):890–896, 2003.

[UBZ06] D. Kundur U. Budhia and T. Zourntos. Digital video steganalysis exploiting statistical visibility in the temporal domain. *IEEE Trans. on Information Forensics and Security*, 1(4):502–516, 2006.

[vqm96] *American National Standard for Telecommunications-Digital Transport of One-Way Video Signals-Parameters for Objective Performance Assessment*. American National Standards Institute, Alliance for Telecommunications Industry Solutions, 1996.

[Wat93] A. Watson. DCT quantization matrices visually optimized for individual images. *SPIE*, 1913:202–216, 1993.

[WB94] S. Wicker and V. K. Bhargava. *Reed-Solomon Codes and Their Applications*. IEEE, 1994.

[WCC08] F. Autrusseau W. Chen and P. Callet. A robust watermarking algorithm for H.264/AVC in compressed domain. In *ECN and KEIO Global COE Joint Workshop*, pages 120–124, 2008.

[WGE03] S. Winkler, E. Gelasca, and T. Ebrahimi. Toward perceptual metrics for video watermark evaluation. In *in Proc. of SPIE, Applications of Digital Image Processing*, pages 371–378, 2003.

[WS03] T. Wiegand and G. Sullivan. *Advanced video coding for generic audiovisual services*. International Tellecommunication Union, 2003.

[WZZ00] R. Wang, H. Zhang, and Y. Zhang. A confidence measure based moving object extraction system built for compressed domain. In *International Symposium on Circuits and Systems*, volume 5, pages 21–24, 2000.

[XWW11] D. Xu, R. Wang, and J. Wang. A novel watermarking scheme for H.264/AVC video authentication. *Signal Processing: Image Communication*, 26(6):267–279, 2011.

[XYH+] Z. Xiao, C. Yong, C. Hui, Z. Shuo, and X. Zhang. Drift Compensation in Compressed Video Reversible Watermarking,. In *World Congress on Computer Science and Information Engineering*.

[YBD04] C. Laurent Y. Bodo, N. Laurent and J. Dugelay. Video waterscrambling: Towards a video protection scheme based on the disturbance of motion vectors. *EURASIP Journal on Applied Signal Processing*, 14:22242237, 2004.

[ZB09] D. Zou and J. A. Bloom. H.264/AVC substitution watermarking: a CAVLC example. In *Media Forensics and Security*, pages 725–740, 2009.

[ZB10] D. Zou and J. Bloom. H.264 stream replacement watermarking with CABAC encoding. In *International Conference on Multimedia & Expo*, pages 117–121, 2010.

[ZH06] J. Zhang and A. T. S. Ho. Efficient Video Authentication for H.264/AVC. In *International Conference on Innovative Computing, Information and Control*, volume 3, pages 46–49, 2006.

[ZHQM07] J. Zhang, A. Ho, G. Qiu, and P. Marziliano. Robust Video Watermarking of H.264/AVC. *IEEE Trans. on Circuits and Systems II: Express Briefs*, 54(2):205–209, 2007.

[ZLZ01] Jun Zhang, Jiegu Li, and Ling Zhang. Video watermark technique in motion vector. In *Computer Graphics and Image Processing, 2001 Proceedings of XIV Brazilian Symposium on*, pages 179–182, Oct 2001.

[ZWH04] L.-H. Zhang, H.-T. Wu, and C.-L. Hu. A video watermarking algorithm based on 3D Gabor transform. *Journal of Software*, 14(8):12521258, 2004.

[ZZP10] L. Zhang, Y. Zhu, and L. Po. A novel watermarking scheme with compensation in bit-stream domain for H.264/AVC. In *ICASSP*, pages 1758–1761, 2010.

Publications Related to Dissertation

- T. Dutta, A. Sur, S. Nandi, "Motion Coherent Region Detection in H.264 Compressed Videos," 2013 IEEE International Conference on Multimedia and Expo (ICME) 2013, San Jose, California, USA.

- T. Dutta, A. Sur, S. Nandi, "A Robust Compressed Domain Video Watermarking in P-frames with Controlled Bit Increase Rate," 9th annual National Conference on Communications (NCC) 2013, Delhi, India, pp.1-5.

- T. Dutta, "Motion Compensated Compressed Domain Watermarking," Doctoral Symposium in ACM Multimedia (ACM MM) 2013, Barcelona, Spain, pp.1039-1042.

- T. Dutta, A. Sur, S. Nandi, "Compressed Domain Robust Video Watermarking," Microsoft TechVista, Microsoft Research India, Coimbatore, January 2013.

Brief Biography of the Author

Tanima Dutta was born in Durgapur, West Bengal, India on 27th January, 1983. After completing her schooling in Durgapur, she has completed the B.Tech. degree from the Department of Computer Science and Engineering, Haldia Institute of Technology, West Bengal, India in the year 2005. She has received SAIL Scholarship from Steel Authority of India for pursuing her B.Tech. programme. She completed her M.Tech. degree from the A. K. Choudhury School of Information Technology, Calcutta University, West Bengal, India in 2010. She has pursued her Ph.D. degree in the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, Assam, India under the supervision of Dr. Arijit Sur. She has received TCS Research Fellowship from TATA Consultancy Services, India for pursuing her Ph.D. programme. Her research interests include multimedia security, computer vision, machine learning etc. She has also worked in other fields, such as multimedia in wireless sensor network, medical imaging, etc.