AN IMPROVED SCALAR QUANTIZATION-BASED DIGITAL VIDEO WATERMARKING SCHEME FOR H.264/AVC

by

ADARSH GOLIKERI

B.E., Bangalore University, 2001

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIRMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

October 2005

Abstract

Digital watermarking, a robust information-embedding technique, has gained significant attention in the past few years, due to the spread of illegal redistribution and unauthorized use of digital multimedia content. In general, a watermark is a secure, perceptually invisible, unique, low-power signal which is robustly inserted into original digital content.

In this thesis, we propose an improved, scalar quantization-based digital video watermarking scheme. The aim is to enable video content producers and owners to embed a robust watermark into their video. If such a scheme is implemented on a large scale, it could serve as a deterrent against rampant distribution and sharing of pirated copies of video content. Our scheme embeds a locally adaptive, robust, Rate-Distortion (R-D) optimized watermark signal into the transform domain of the macroblock residual. This ensures that watermark signal is embedded in the most robust manner, with least visual distortion. We use a unique perceptual mask which limits the amount of spatial and temporal distortion due to watermark insertion. Therefore, our scheme achieves higher watermarked picture quality compared to existing schemes. Our scheme is designed with a built-in bit-rate controller, which ensures that the watermark bits are distributed in proportion to the visual importance of different regions of the video frame.

We adapt our scheme to H.264/AVC, which is the latest video coding standard. Our scheme overcomes the challenges for watermarking of H.264/AVC video, namely high compression efficiency, small residual data, integer transform, R-D coding decisions and video bit-rate control. Experimental results on several standard video sequence show that compared to existing quantization-based watermarking schemes, our proposed scheme is significantly more robust in terms of Bit Error Rate (BER) to different types of attacks, including video compression and decompression, transcoding, low-pass filtering, scaling, rotation and collusion.

Table of Contents

List of Tables

List of Figures

Acknowledgements

I would like to thank my supervisors Dr. Panos Nasiopoulos and Dr. Z. Jane Wang for their constant support, guidance and encouragement through the entire duration of my Master's degree at UBC. I also owe a lot to the students and faculty associated with the Image Processing Laboratory at UBC. In particular, I would like to extend my heartfelt thanks to Dr. Rabab Ward, who always put the interest of her students ahead of anything else. My research colleagues Hassan Mansour and Lino Coria have been a great source of creative and fruitful discussions. A special thanks goes to Mehrdad Fatourechi, our Lab administrator, who took great pains to ensure that the lab resources were always up and running. My friends at UBC were a great support through the duration of my Master's and I am grateful to them. Finally, I would like to thank my family back home for their constant encouragement, love and support. I would like to dedicate this thesis to the memory of my late grand father Mangesh Ganapatrao Golikeri, who still remains my strongest inspiration.

1 Introduction

1.1 Thesis Objective

The last decade has witnessed an enormous growth of the demand for digital multimedia content. In the last decade, technologies such as Digital Versatile Disk (DVD) for video content and MPEG Layer 3 (MP3) for audio content have been introduced for the benefit of the consumer, since they provided a very high quality with very little or no degradation (as opposed to analog VHS or magnetic audio tapes). However, in the past few years, there has been an increase in the illegal redistribution and unauthorized use of digital multimedia. Today, content owners and producers are concerned about the lack of proper protection mechanisms for their data. Traditional mechanisms such as encryption can only protect the data to the point when it is decrypted and presented to the end user.

Digital watermarking has attracted a great deal of research interest as a strong, complementary technology that can protect content even *after* it has been decrypted ([1], [2], [3], [4]). Watermarking is an information-embedding technique by which a *secret*, *imperceptible* signal – a watermark – is embedded directly into the original digital content (also called host signal or Cover Work) in a *robust* manner. This watermark is designed to survive a wide range of common signal processing distortions such as compression, filtering, digital-to-analog conversion, as well as malicious attacks such as collusion [2].

Scalar Costa Scheme (SCS) has been shown to be a reliable, scalar quantization-based information-embedding technique [5]. It outperforms the popular Spread Spectrum

(SS) watermarking techniques, due to its host-interference rejecting properties. However, SCS is a generic framework and has certain limitations which prohibit the direct use of SCS for video watermarking.

In this thesis, we propose an improved SCS for digital video watermarking scheme which removes the limitations of the traditional SCS. Our scheme embeds a locally adaptive watermark based on the host signal characteristics. We use Rate-Distortion (R-D) coding to ensure an optimum watermark signal. We also propose a simple but effective perceptual mask which controls the level of spatial and temporal distortions in the watermarked video. Our scheme is designed with a built-in bit-rate controller in order to ensure optimum watermark bit allocation. Our scheme is then adapted to H.264/AVC (Advanced Video Coding), which is the latest video coding standard of the ITU Telecommunication Standardization Sector (ITU-T) and International Organization for Standardization (ISO) / International Electrotechnical Commission (IEC). Our scheme is easy to integrate into the existing H.264/AVC encoder and decoder and can operate in real-time. It does not require transmission of any overhead data. We compare our scheme with the traditional SCS and show significant improvement in robustness against several different attacks – H.264 compression and decompression, transcoding, filtering, scaling, rotation and collusion.

1.2 Thesis Outline

This thesis is organized as follows. In the remaining part of this chapter, we provide an overview of several essential concepts in video watermarking, necessary to comprehend the later chapters of this thesis. In Section 1.3, we provide an overview of digital watermarking, with an emphasis on techniques for video watermarking and its applications. In Section 1.4, we present an overview of the existing Scalar Costa Scheme for information embedding. We also discuss the performance and limitations of this scheme.

In Chapter 2, we present our proposed digital video watermarking scheme. Section 2.1 describes our watermark encoder in detail. We discuss the new features of our scheme, which include the locally adaptive watermark embedding strength, R-D based watermark embedding, derivation of the spatial and temporal perceptual masks and the bit-rate controller. We discuss how our scheme can be integrated into an existing video encoder. In Section 2.2, we provide a description of the watermark decoder used in our scheme.

Then, we adapt our scheme to the specific case of H.264/AVC video in Chapter 3. In Section 3.1, we examine certain features of H.264/AVC that are relevant to our proposed scheme. This includes the Intra (I-) and Inter (P-) prediction modes, Transform, Quantization and R-D Optimized video coding. We also present challenges in H.264/AVC watermarking and solutions to these challenges. In Section 3.2, we consider the design issues for adapting our scheme to H.264/AVC. This includes the selection of the Watermark Quantizer value, selection of transform coefficients for watermarking and the Lagrangian parameter for R-D optimized watermark embedding.

Finally, we present experimental results in Chapter 4. In Section 4.1, we discuss details of the implementation of our scheme on various standard video sequences. Section 4.2 details the various robustness tests for our scheme and traditional SCS. Video quality measures are presented for each test. We then present conclusions and contributions in Chapter 5.

1.3 Overview of Digital Watermarking

The history of watermarking can be traced back to the late Thirteenth century, when paper watermarks were used in Italy [1]. The watermark was a thin wire pattern embedded inside the paper. When held up to the light, these marks resembled the effect of water on paper and hence the term *watermark* was used to describe them. A common example of watermarking is a currency note which has several security features embedded in it. Just as the main purpose of watermarking a currency note is to avoid counterfeiting, the aim of Digital Watermarking is to protect the copyrights of digital content.

1.3.1 Fundamentals of Digital Watermarking

Digital Watermarking can be defined as the process of *robustly* embedding a *secret*, *imperceptible* signal directly into the host signal. The host signal is often called the Cover Work, or simply work. The watermark should be resistant to both malicious and unintentional attacks. Malicious attacks are those in which two or more users in possession of valid watermarked copies may collude to produce a new copy which is unwatermarked, or a copy with a new, valid watermark. Unintentional attacks can include a wide range of signal processing distortions – low pass filtering, denoising, geometric distortions (scaling, rotation, shearing, etc.), Digital-to-Analog (D/A) conversion, valumetric scaling (brightness and/or contrast changes), recompression and transcoding. It is important to note that the watermark need not be resistant to *all* possible attacks. We consider only the subset of attacks which preserve the *perceptual meaning* of the host signal. The watermark detection should gracefully degrade with the quality of the host signal.

There is a complex trade-off between 3 parameters in digital watermarking – payload, fidelity and robustness. Payload is the number of watermark bits that can be embedded in a given host signal. Fidelity refers to the distortion due to watermark embedding. Robustness is the resistance to attacks. Embedding a strong watermark will make it more robust, but will result generally in poor fidelity of the watermarked work. Also, if the payload of the watermark is very high, it often leads to poor robustness since there is a high probability that the attacks will affect a larger number of watermark bits [2].

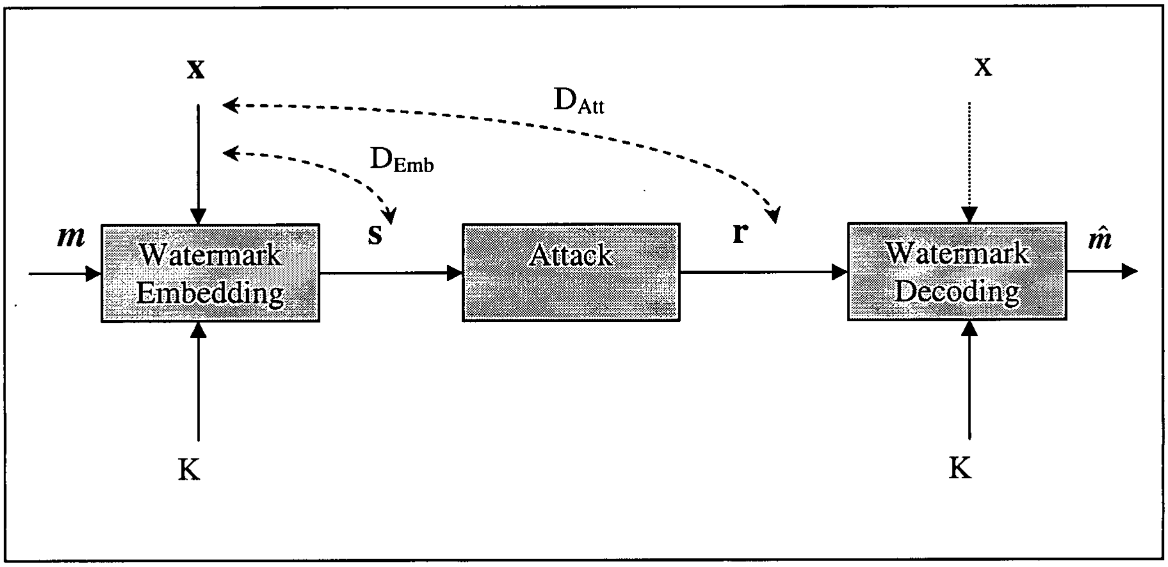


Figure 1.1 General model of digital watermarking

In Figure 1.1, the general model of digital watermarking is shown. In this thesis, **bold** text is used to denote a vector while plain text and *italics* are used to denote scalar quantities. The host signal **x** is embedded with a watermark message *m*, with the use of a secure key K. The watermarked work **s** has a distortion D_{Emb} compared to **x**. The watermarked signal then may undergo an attack, resulting in an attacked work **r**, with distortion D_{Att}. At the watermark decoder, the original, unwatermarked work **x** may or may not be available. These two scenarios are termed non-blind and blind watermark detection, respectively. With the aid of the secure key K, an *estimate* $\hat{m}$ of the watermark message m is extracted [5].

Some common terms that will be used in this thesis are explained now. The terms *private data* for the original data and *public data* for the watermarked data are commonly used. Also, the terms *public-key* and *private-key* watermarking are used to distinguish between systems where the watermarking key is publicly available or limited to a small group of users (e.g. copyright holders). The watermarking schemes considered in this thesis are all *private-key* schemes. Sometimes, the term *informed watermarking* is used to indicate that the original data is available. If the watermark embedder exploits information about the original data while inserting the watermark, it is called *informed embedding*. Our scheme belongs to this class of watermark systems. Another common term for informed embedding is *communication with side information at the encoder* ([5],[6],[7],[8]). The term *attack* includes any processing which alters the watermarked data in some way, which may have an effect on the watermark decoding process. Generally, watermarking algorithms are designed to survive distortion up to a particular level, say D_{Att}.

1.3.2 Applications of Watermarking

In Table 1.1, the most common applications of watermarking are listed. The focus here is on video watermarking, but some of these applications may be extended to other multimedia data as well. Digital fingerprinting refers to the process in which a unique watermark is embedded into each licensed, legal copy of a work so that when an illegal copy is found, the traitor can be identified and sued in court. This application is very relevant in the present day scenario, especially for digital video and audio content. The explosion of the internet and specifically the rapid, uncontrolled growth of peer-to-peer file sharing mechanisms (KaZaa, Morpheus, eDonkey, Gnutella, etc.) has lead to a situation wherein popular Hollywood movies and music albums are often available for free download close to their release date [9]. As a result of this, copyright holders (movies studios, recording studios, artists, etc.) stand to lose a significant amount of revenue. It must be pointed out that the basic problem is not with the internet or peer-to-peer networks themselves, but a traitor who has made digital copies of the content available for free, illegal download. Thus, watermarking can be very useful in identifying the source of piracy. This kind of mechanism will definitely act as a deterrent against piracy.

Table 1.1 APPLICATIONS OF DIGITAL WATERMARKING

Applications	Purpose
Digital Fingerprinting	To trace the source of pirated data
Data Authentication	To verify the genuineness of data
Copyright protection	To prove ownership of data
Copy control	To prevent unauthorized coping of data
Broadcast monitoring	To verify broadcast content
Video coding enhancements	To provide supplemental information (E.g. for Error concealment)

Data authentication is an important application of watermarking, due to the ease with which digital content can be edited and manipulated. For example, if the video from a surveillance camera is used as evidence in a court of law, the authenticity of the content has to be verified and this system has to be fool-proof. In this case, the watermarking scheme is used to embed information about the distinctive features of the content (e.g. the edge map of a video frame). So, if the content has been modified in anyway, the detector will find a discrepancy between the features extracted from the altered content and those extracted from the watermark information [2].

Copyright protection involves inserting a watermark which contains information about the content owner. In case an illegal copy is found, then the owner can prove his identity by extracting the watermark and use it as evidence. However, this approach is not without certain drawbacks. If a malicious user embeds his own watermark into the content, he too can claim ownership. To solve this problem, watermarks for copyright protection are generally non-blind i.e. they require the original content to be present for verification. Since only the legal content owner will have access to the unwatermarked content, the ownership issue can be resolved. This requires the watermarking scheme to be non-invertible. Also, such schemes are generally backed up by a third party. The copy protection system for the DVD standard was proposed in [10].

Watermarking can also be used a part of a larger Copy control mechanism. In this application, the recording capabilities of a digital device are limited or controlled by the information present in the source content. The Content Scrambling System (CSS) for DVD is one such example. CSS scrambles the video recorded onto a DVD. For descrambling, a pair of keys are required – one is stored in the lead-in area of the disk and the other is stored in the MPEG video file. There is also the Copy Generation Management System (CGMS), which is a pair of bits stored in the MPEG stream header, with one of the three possible rules for copying: *copy-always*, *copy-never* and *copy-once.* In this case, watermarking is used to protect the CGMS bits because they do not survive Digital-to-Analog conversion.

Broadcast monitoring involves watermarking broadcast content such as paid advertisements, so that content owners get paid correctly and advertisers get what they paid for. The whole television market is worth several billions of dollars and Intellectual Property (IP) violations are bound to occur. For example, the value of a 30 second commercial during the 2002 FIFA World cup was around $120,000. Thus, it is very essential to have an automated broadcast surveillance system setup. A system called *active monitoring* has been designed, wherein a real-time watermarking scheme transmits identification information along with the video data. This allows for simultaneous monitoring of many channels, without the need for human intervention.

A less explored application of watermarking is for enhanced video coding. Watermarking can be used in Error concealment for video. In this case, there is no need to transmit any redundant information along with the video stream. Experiments have shown that such a mechanism can even outperform traditional error concealment schemes. Another interesting area is the use of watermarking in hiding one video stream inside another (Picture-in-Picture systems).

1.4 Video Watermarking

Research on Digital Watermarking has focused mainly on images. However, such schemes cannot always be directly extended to video. This is because video coding has its own peculiarities. Three of the main challenges in video watermarking [9] are discussed below:

1.4.1 Common Video processing

There are a wide range of video editing and processing tools available (VirtualDub, AviSnyth, etc.) today. These tools can be used by anyone from the content creator to the end-user, in order to suit their own needs. From a watermarking point-of-view, this means that the watermark has to be resistant to all such processing, as long as the perceptual quality of the video is retained.

Table 1.2 COMMON VIDEO PROCESSING EXAMPLES

Video processing	Example
Video Editing	▪ Fade, Dissolve, Wipe ▪ Subtitles, Logo overlay
Desynchronization (spatial)	▪ Aspect ratio changes ▪ Jitter ▪ Spatial resolution change
Desynchronization (temporal)	▪ Frame rate conversion
Photometric	▪ Brightness / contrast ▪ Gamma correction ▪ Unsharp masking ▪ Spatial Smoothing / blurring ▪ Temporal smoothing ▪ Denoising ▪ Histogram Equalize / Stretch
Geometric	▪ Resize ▪ Rotate
Transcoding	▪ Format conversion (H.264, MPEG-2, DivX, WMV-9, MOV) ▪ GOP structure change

Table 1.2 lists several categories of processing and also examples for each. Using a video editing suite, a user may want to add certain visually pleasing transitions to the video content, such as a fade or a dissolve. He might also want to overlay a graphic or a logo over the video. Spatial desynchronization includes aspect ratio changes (e.g. 16:9 to 4:3 conversion) and spatial resolution changes (e.g. NTSC – PAL conversion). Another attack is positional jitter, which occurs if the video in a cinema theater has been captured by a handheld camera held at a misaligned angle. A common temporal desynchronization

attack is frame-rate conversion. Such a conversion would affect watermarking schemes which use a different key K for each frame. Photometric attacks are the largest category and also the most frequently used. Sometimes, it may be necessary to apply video filters in order to mitigate the effect of signal noise. Gaussian blurring (smoothing), unsharp masking and denoising are examples. In other cases, brightness, contrast or gamma correction might be necessary for certain frames. Geometric attacks include the commonly used resize operation (to reduce video file size) and the less common rotate operation [9].

The advent of several popular video codecs has created another challenge for watermarking – transcoding. A pirate generally will transcode the source video (say a DVD in MPEG-2 format) into a more recent or advanced format (such as H.264, DivX, WMV-9 or Real Media) in order to substantially reduce the amount of data. For example, a DVD movie which normally takes up about 4.7GB of data, can easily be compressed using a standard PC and free conversion tools (E.g. DVD Decrypt, Gordian Knot, AutoGK), resulting in a video file of about 700MB ! The entire process would take no longer than a few hours and the resulting video quality is *almost* as good as the original DVD. This presents a great challenge to watermarking schemes. If a watermark is able to survive the transcoding operation, it would definitely help movie studios to trace the source of piracy through Digital Fingerprinting.

Another example of transcoding is changing the Group-of-Pictures (GOP) structure of a video stream at the same bit-rate. For example, if the original video had one Intra-frame (I-frame) every 5 seconds, an attacker could transcode it to produce a stream which has an I-frame every 12 seconds. Although the resulting video would be indistinguishable from the original, it changes the video prediction and residual data considerably. This could work against certain watermarking schemes. In Chapter 4, we show that our proposed scheme is resistant to a wide range of video processing attacks.

1.4.2 Real-time Constraints

Watermarking of still images does not really require the scheme to function in real-time. However, in video watermarking even a few seconds of delay per frame of video is unacceptable. This is because video is generally transmitted at a high frame-rate (25 frames-per-second) to avoid flicker. Similarly, in broadcast monitoring, the watermark detection should be in real-time. This puts a limit on the complexity of the watermark embedder and decoder. It should be noted that if the watermark operates in the compressed domain (such as transform coefficients) rather than the uncompressed domain (spatial pixel values), the time requires for watermarking can be significantly reduced. For example, several watermarking schemes alter the Variable Length Code words (VLC) of a video stream to embed information. Another option to reduce real-time constraints is to split the watermarking process into two steps – pre-processing and embedding. In the pre-processing step, the watermark embedder analyses the video stream and computes the appropriate watermark signal. Next, the embedder inserts the watermark signal during the video encoding step without any delay. In Chapter 3, we discuss the real-time operating characteristics of our watermarking scheme.

1.4.3 Collusion resistance

The challenge of collusion for video watermarking is bigger than that for images, because of the availability of both spatial and temporal dimensions ([11], [12]). Collusion refers to a group of malicious users who utilize different watermarked content in order to create illegal content i.e. unwatermarked content. There are two main types of collusion to be considered:

Collusion Type I – When the *same* watermark is inserted into *different* copies of *different* video content, pirates can estimate the watermark from these copies and use this knowledge to obtain unwatermarked video content (Figure 1.2). This is generally the case in Copyright protection. The estimate can be obtained from the fact that watermarks generally resemble noise. Hence, if the watermarked content is subtracted from a low-pass filtered version of itself, a simple estimate can be obtained. Once an accurate estimate is obtained from the different copies, the unwatermarked copy is generated by simply subtracting the watermark from the content.

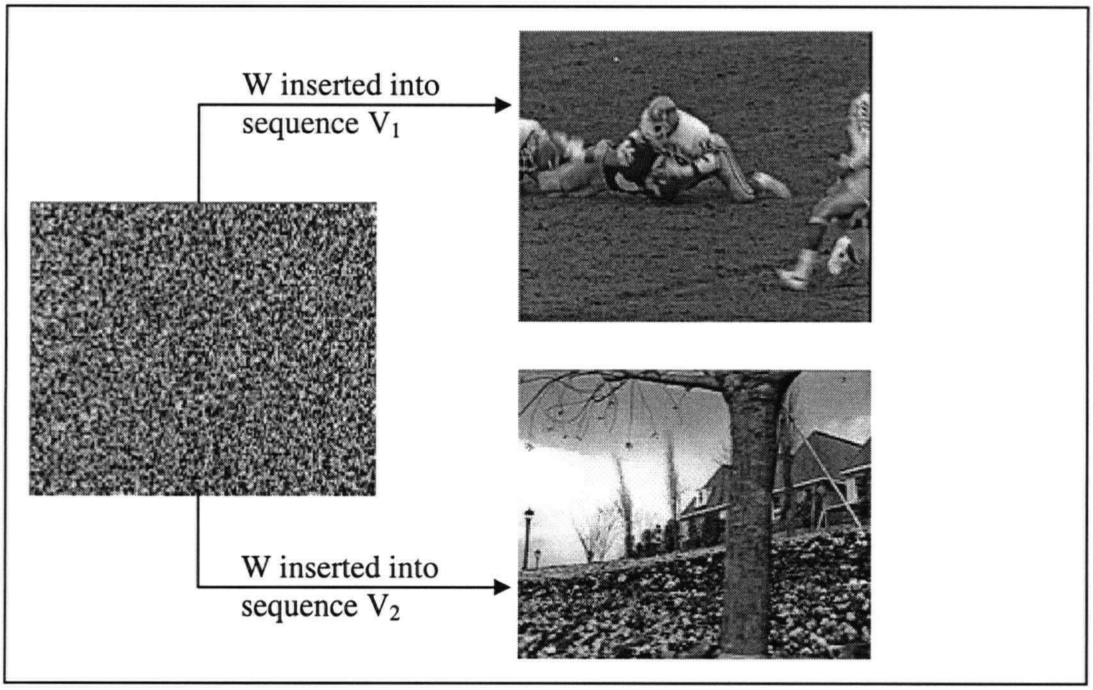

W inserted into sequence V_1
W inserted into sequence V_2

Figure 1.2 Scenario for Collusion Type I

Collusion Type II – When *different* watermarks are inserted into *different* copies of the *same* video content (Figure 1.3). This is generally the case in Digital Fingerprinting. A simple example is linear collusion, where several legitimate, watermarked copies are averaged in order to generate a new unwatermarked work. The strength of the watermark diminishes with an increase in the number of watermarked copies available to the malicious user.

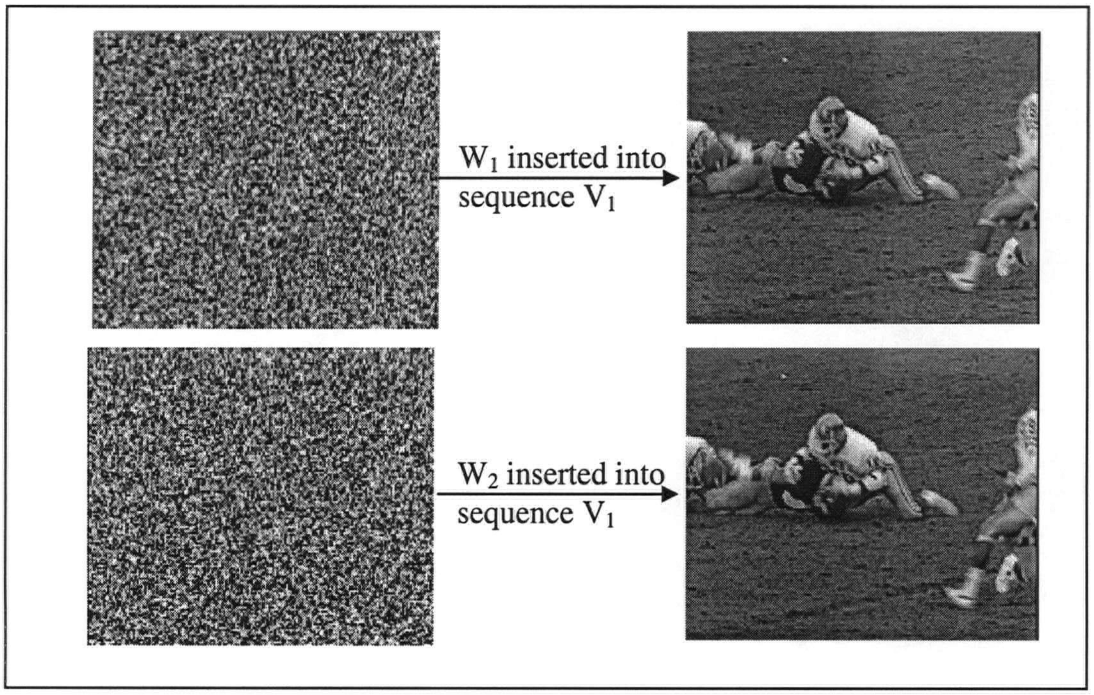

W_1 inserted into sequence V_1
W_2 inserted into sequence V_1

Figure 1.3 Scenario for Collusion Type II

Intra-video Collusion – This type of collusion is unique to video. If the same watermark is inserted in each frame of a video sequence, collusion type I can be applied, because accurate estimates of the watermark can be obtained by simply analyzing each individual frame. If different watermarks are embedded for each frame, then the watermark can be diminished by averaging those frames which have little or no motion between them (static frames). Therefore, in both cases, it is possible for a malicious user to work on one single watermarked video sequence in order to remove the watermark.

To counter this situation, a basic rule has been proposed in [9] for video watermarking:

- If two frames are similar, then the watermarks inserted into them should be highly correlated.
- If two frames are different, then the watermarks should be highly uncorrelated.

In fact, this is the basic principle of informed watermarking. The idea is to have a host signal dependent watermark. In Chapter 2, we discuss the locally adaptive watermark of our proposed scheme, which varies according to the content of the video. Thus, our scheme is inherently robust against collusion attacks.

1.5 Overview of the Scalar Costa Scheme (SCS)

A block diagram of a typical blind watermarking scenario is shown in Figure 1.4 [7]. The watermarking process can be considered as communication with *side-information at the encoder* ([2], [3]). Using a secure key K, the watermark message *m* is embedded into the cover work **x** (which is modeled as independent identically distributed (IID) data) of variance σ_x^2. The watermark is defined as $\mathbf{w} = \mathbf{s} - \mathbf{x}$ and has a variance σ_w^2. The embedding distortion D_{Emb} is defined as the mean-squared error between **s** and **x**,

$$D_{Emb} = \frac{1}{n} E\left\{\|\mathbf{s} - \mathbf{x}\|^2\right\}, \tag{1.1}$$

where ||.|| is the Euclidean norm operator and n denotes the size of **s** and **x**. The watermarked signal **s** is then transmitted over a channel which introduces an additive white Gaussian noise (AWGN) **v** of variance σ_v^2, resulting in an attacked work **r**. The decoder receives **r** and extracts the watermark message estimate m, using the same key K which was used during embedding. The mapping of m onto the sequence **w** is determined by **x** and by the codebook *W(K)*, which is encrypted by the key K. In watermarking, it is generally assumed that the watermark sequences **w** have zero mean and unit variance. Therefore, $D_{Emb} = \sigma_x^2$. The AWGN attack distortion is D_{Att}.

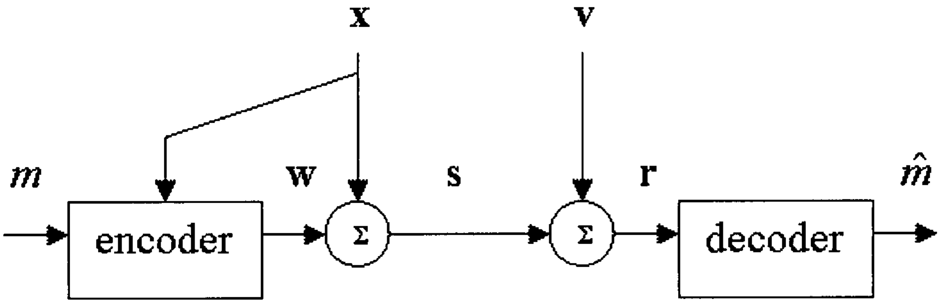


Figure 1.4 Typical blind watermarking scenario

1.5.1 Costa's result

For a discrete memoryless channel, it has been shown in [6] that for the case of communication with side information at the encoder, the capacity is:

$$C_{ICS} = \max_{u}(I(u;r) - I(u;x)) \ , \tag{1.2}$$

where u is an auxiliary random variable. *I(u;r)* and *I(u;x)* represent the mutual information between u and r, and the mutual information between u and x respectively. r is a random variable which denotes received data while x denotes additive channel noise, which is side information to the encoder. ICS denotes Ideal Costa Scheme (or simply

Costa's scheme). In the case of blind watermarking, x denotes the host signal. At the encoder, the signal to be transmitted is determined based on the message m, realizations $\mathbf{u}$ of u for all possible message combinations and the side information $\mathbf{x}$ which is available with the encoder. These realizations $\mathbf{u}$ are stored in a codebook U, which is known to both the encoder and decoder. Costa [5] proposed a solution to the communication problem shown in Figure 1.5. The main ingredient was the design of an n-dimensional codebook U^n. In the limit as $n \rightarrow \infty$, Costa's codebook achieves the capacity of communication with IID Gaussian side information $\mathbf{x}$ at the encoder and an AWGN channel. Costa defines his codebook as:

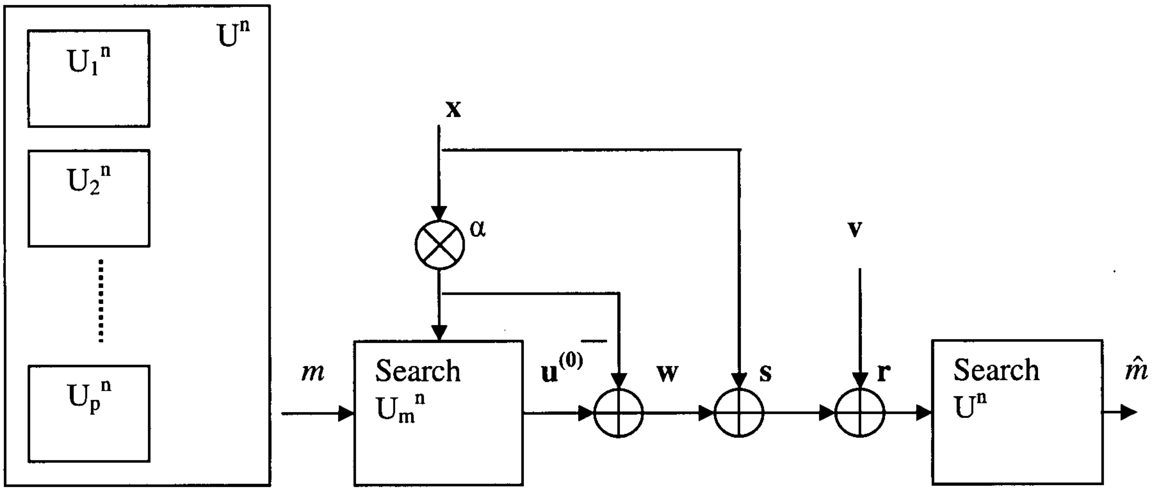


Figure 1.5 Structure of Costa's scheme

$$U = \{u_l = w_l + \alpha x_l \mid l \in \{1,2,...n_u\}\}, \tag{1.3}$$

$$\mathbf{w} \sim N(0, \sigma_w^2 I_n), \mathbf{x} \sim N(0, \sigma_x^2 I_n),$$

where $\mathbf{w}$ and $\mathbf{x}$ are realizations of two n-dimensional independent random processes w and x, with Gaussian PDF and I_n is the n-dimensional identity matrix and α is the codebook parameter, with $0 \leq \alpha \leq 1$. The number of codebook entries is given by

$$n_u = \text{ceil}\left(2^{\,n.I(u;r)-\varepsilon} \right), \qquad (1.4)$$

where ε denotes an arbitrarily small positive value and ceil(.) denotes rounding to the next largest integer. The codebook is partitioned into p disjoint codebooks in such a way that each sub-codebook U_p^n contains the same number of sequences. Thus, the total codebook is denoted as $U^n = U_1^n \cup U_2^n \cup \ldots\ldots U_p^n$. The encoding process works as follows: First, a pair $(\mathbf{u}^{(0)},\mathbf{x})$ in the sub-codebook U_p^n is found. A precise derivation of jointly typical sequences is out of the scope of this thesis. But it is sufficient for this work to consider finding the codebook entry $\mathbf{u}^{(0)}$ such that $\mathbf{w} = \mathbf{u}^{(0)} - \boldsymbol{\alpha}\,\mathbf{x}$ is nearly orthogonal to $\mathbf{x}$. Second, the watermarked data is given by $\mathbf{s} = \mathbf{x} + \mathbf{w}$. A fundamental difference with the traditional Spread Spectrum (SS) approach can be noted here. The codebook of all possible watermark sequences **w** is *infinite*. So, an appropriate watermark sequence **w** is derived from an entry in the auxiliary codebook U which has a finite number of entries and the given host signal **x**.

The watermark decoder receives $\mathbf{r} = \mathbf{w} + \mathbf{x} + \mathbf{v}$. It then searches the entire codebook for a sequence **u** such that **(u,r)** is jointly typical. There is a high probability that this sequence is $\mathbf{u}^{(0)}$. The index $\hat{m}$ of the sub-codebook containing **u** is the decoded watermark message. Costa showed in [6] that for the codebook in **(1.3)** with:

$$\alpha = \frac{\sigma_w^2}{\sigma_w^2 + \sigma_v^2} = \frac{1}{1 + 10^{-WNR/10}} , \qquad \mathbf{(1.5)}$$

the capacity is given by:

$$C_{ICS}^{AWGN} = \frac{1}{2}\log_2\left(1 + \frac{\sigma_w^2}{\sigma_v^2}\right) , \qquad \mathbf{(1.6)}$$

where WNR denotes the Watermark-to-Noise Ratio (in dB) = $10*\log10(\sigma_w^2/\sigma_x^2)$. Costa's most important observation was **(1.6),** which tells us that the capacity is completely independent of knowledge of the original data **x** and σ_x^2. However, it must be noted that ICS is not a practical scheme, because the size n_u of the codebook U can become very large even for moderately large data length *n.* Besides, there is also the additional problem of storing and searching the codebook U due to its huge size and random structure.

1.5.2 The Scalar Costa Scheme (SCS)

In order to use Costa's scheme, one needs to have an infinitely large, random codebook. From a practical point of view, the codebook should be as small and structured as possible. Therefore, in [5], a suboptimal, practical information embedding scheme based on Costa's result is proposed, called Scalar Costa Scheme (SCS).

To obtain a structured codebook, U is chosen to be a product codebook of dithered uniform scalar quantizers. First, the watermark message *m* is converted into a binary representation **b.** Then, **b** is encoded into a sequence of watermark letters **d**. The elements d_n belong to a D-ary alphabet, D = {0,1...D-1}. Throughout this thesis, binary SCS watermarking is considered ($d_n \in$ D = {0,1}). Second, U^n which is the *n*-dimensional codebook of ICS, is structured as a product codebook $U^n = U^1 \bullet U^1 \bullet \ldots\ldots \bullet U^1$ of *n* 1-dimensional codebooks U^1, where all component codebooks are identical. For D-ary signaling, the component codebook U^1 is separated into D disjointed parts:

$$U^1 = U_0^1 \cup U_1^1 \ldots\ldots \cup U_{D-1}^1 . \quad \textbf{(1.7)}$$

In SCS, the codebook U^1 is constructed as a scalar uniform quantizer of step Δ:

$$U^1(\alpha,\Delta,D) = \{\ u = l\alpha\Delta + d\ (\alpha\Delta / D)\ |\ l \in Z, d \in D\ \}, \quad \textbf{(1.8)}$$

where Z denotes the set of integers, l enumerates all quantizer representatives of a scalar quantizer with step size $\alpha\Delta$ and d represents a shift of the quantizer. Now, in watermarking, *security of the codebook* is an important issue, which is not handled by regular ICS. Therefore, the authors introduce a secure, pseudorandom key **k** which is derived from the watermark key, with $k_n \in [0,1)$. Therefore, **(1.8)** is modified as follows:

$$U^1(\alpha,\Delta,D,k_n) = \{\, u_n = (l + k_n)\alpha\Delta + d_n\,(\alpha\Delta / D) \mid l \in Z,\ d_n \in D \,\}, \qquad \textbf{(1.9)}$$

where $U^1(\alpha,\Delta,D,k_n)$ is a pseudo-randomly shifted version of $U^1(\alpha,\Delta,D)$. Therefore, an attacker cannot reconstruct the codebook $U^1(K)$ without knowledge of the watermark key K. In order to have a Costa-type information embedding, a jointly typical pair $(\mathbf{u}^{(0)},\mathbf{x})$ has to be found. This is equivalent to finding a sequence $\mathbf{q} = \mathbf{w}/\alpha = (\mathbf{u}^{(0)}/\alpha) - \mathbf{x}$, which is nearly orthogonal to **x** [3]. This search can be considered to be a sample-wise quantization of **x**:

$$q_n = Q_\Delta\left\{x_n - \Delta\left(\frac{d_n}{D} + k_n\right)\right\} - \left(x_n - \Delta\left(\frac{d_n}{D} + k_n\right)\right), \qquad \textbf{(1.10)}$$

where q_n, x_n, d_n and k_n are the elements of the vectors **q**, **x**, **d** and **k** respectively. $Q_\Delta\{\cdot\}$ denotes scalar uniform quantization with step size Δ. Then, the transmitted watermark sequence is:

$$\mathbf{w} = \alpha\mathbf{q}\,, \qquad (1.11)$$

and the watermarked data is:

$$\mathbf{s} = \mathbf{x} + \mathbf{w} = \mathbf{x} + \alpha\mathbf{q}\,. \qquad (1.12)$$

The entire embedding process from **(1.10)** to **(1.12)** is illustrated in Figure 1.6. The embedding of d_n in **(1.10)** is a *subtractive dithered quantization* process, where

$\Delta\left(\frac{d_n}{D}+k_n\right)$ is the dither sequence. The quantization error **q** (and thus also **w**) is almost orthogonal to **x**, assuming uniform original data Probability Density Function (PDF) in the range of one quantization bin. An important property of SCS is that **q** and **w** are statistically independent of **x**. Therefore, SCS can be classified as a host-interference rejecting method of watermarking (as opposed to Spread Spectrum watermarking, which is host-interference non-rejecting).

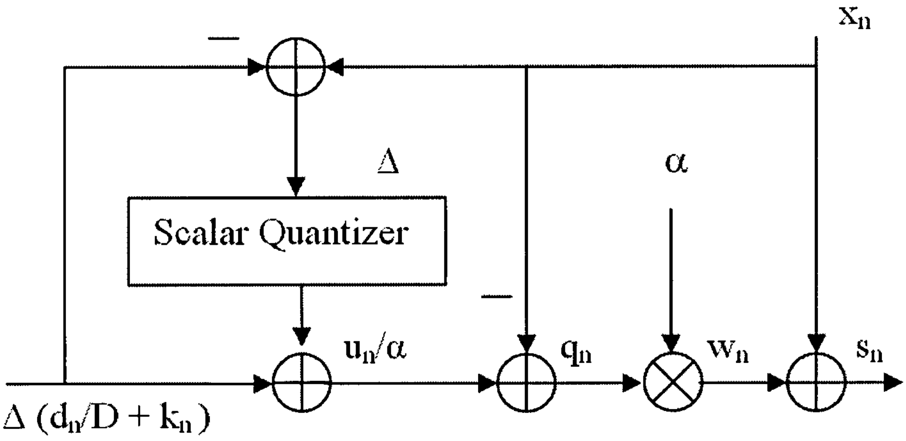


Figure 1.6 Structure of SCS encoder

1.5.3 Watermark Scale Factor α

SCS Embedding depends is entirely dependent on two parameters – quantizer step size Δ and watermark scale factor (or codebook parameter) α. For a given value of watermark power σ_w^2,

$$\alpha = \sigma_w \frac{\sqrt{12}}{\Delta} \quad . \tag{1.13}$$

Now, for ICS, the optimum capacity is obtained by maximizing **(1.2)** over all possible codebooks U. But for SCS, there is only one free codebook parameter α. Thus, the capacity of SCS is given by:

$$C_{SCS} = \max_{\alpha} I(y;d) \ . \tag{1.14}$$

For SCS, it is not possible to compute the maximization over α in **(1.14)** analytically. Thus, the authors optimize numerically for the range of WNRs between -20dB and 20dB.

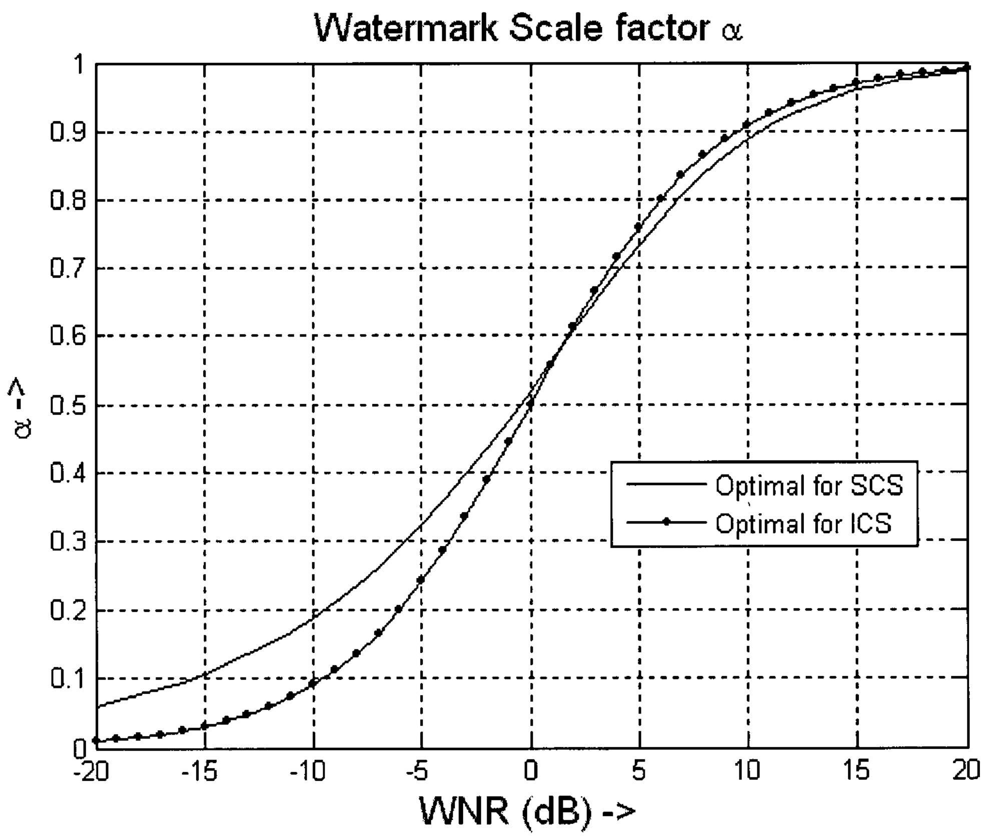


Figure 1.7 Watermark Scale factor α for SCS and ICS vs WNR

An approximate analytical expression for the optimum value of α is derived experimentally as

$$\alpha = \sqrt{\frac{\sigma_w^2}{\sigma_w^2 + 2.71\sigma_v^2}} \ . \tag{1.15}$$

Figure 1.6 shows plots of the optimum α value derived by Costa for ICS **(1.5)** and the optimum α value for SCS **(1.15)** against WNRs in the range of -20 to 20dB.

1.5.4 Limitations of SCS

Before SCS can be used for video watermarking, there are some design issues which have to be solved:

Rate-Distortion Optimization: During video encoding, several coding parameters such as Macroblock prediction modes, motion vectors and transform coefficient quantization levels have to be determined. The problem is compounded by the fact that natural video has widely varying spatial and temporal (motion) content, necessitating the selection of different coding options for different parts of the image. This ensures that the resulting video has a minimum level of distortion for a given bit-rate. Therefore, the task of the video coder is to find a set of coding parameters so that a certain R-D trade-off is achieved for a given decoder. Lagrangian bit-allocation techniques for R-D coding have been widely accepted in recent video codec development due to their effectiveness and simplicity. Thus, it is desirable that the watermark embedding procedure incorporates R-D optimized coding in order to compute the optimum watermark for different regions of a video frame.

Perceptual masking: An important requirement of a watermark is its imperceptibility. This is only possible if an efficient perceptual mask is used during watermark embedding. Now, for video watermarking, both spatial and temporal masking effects have to be considered. SCS does not have any provisions for such perceptual masks. This causes annoying artifacts such as "mosquito effects" in fast-moving regions of the video and blocking artifacts in other regions (Figure 1.8)

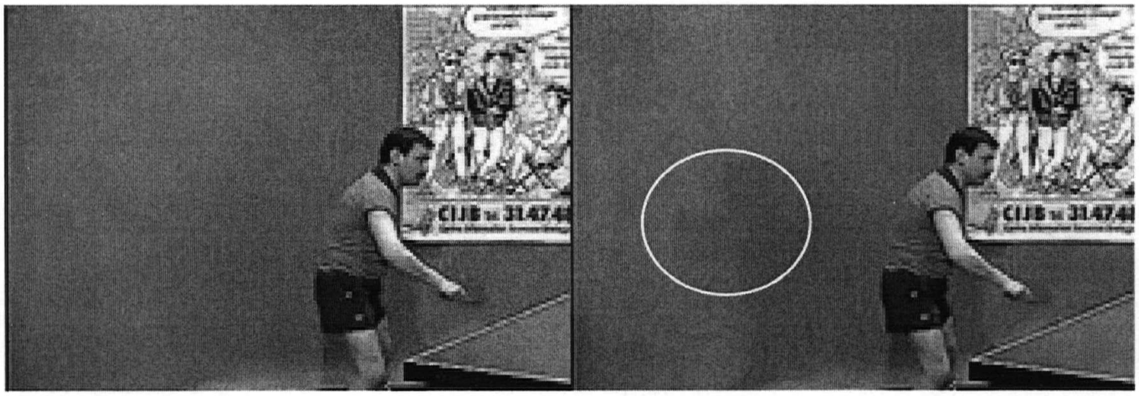


Figure 1.8 Original frame (L) and watermarked frame (R) of Tennis sequence.

Watermark bit allocation: Watermarked video consumes significantly more bits than unwatermarked video. Therefore, it is preferable to have a bit-rate control algorithm in order to trade the fidelity of the watermark with that of the host signal. This algorithm should determine the best allocation of available bits between different watermarked blocks. In video coding, the overall bit-rate is determined by its prediction-mode decisions, motion vector choices and residual signal coding fidelity. Of these, the last factor is most important for bit-rate control. The residual fidelity is controlled by choosing a suitable step-size for quantization of the transform coefficients. For example, H.264 uses a Quantization Parameter (QP) which ranges from 0-51, with 0 representing the least step size and 51 the largest. A larger step size results in lower bit rate and larger distortion. Therefore, the choice of quantization step size is closely related to the relative importance given to rate and distortion. This trade-off is determined by the Lagrangian parameter λ. It has been shown through experimental results that there is a strong relationship between λ and step size. For the case of H.264:

$$\lambda = 0.85 * 2^{(QP-12)/3}. \qquad \textbf{(1.16)}$$

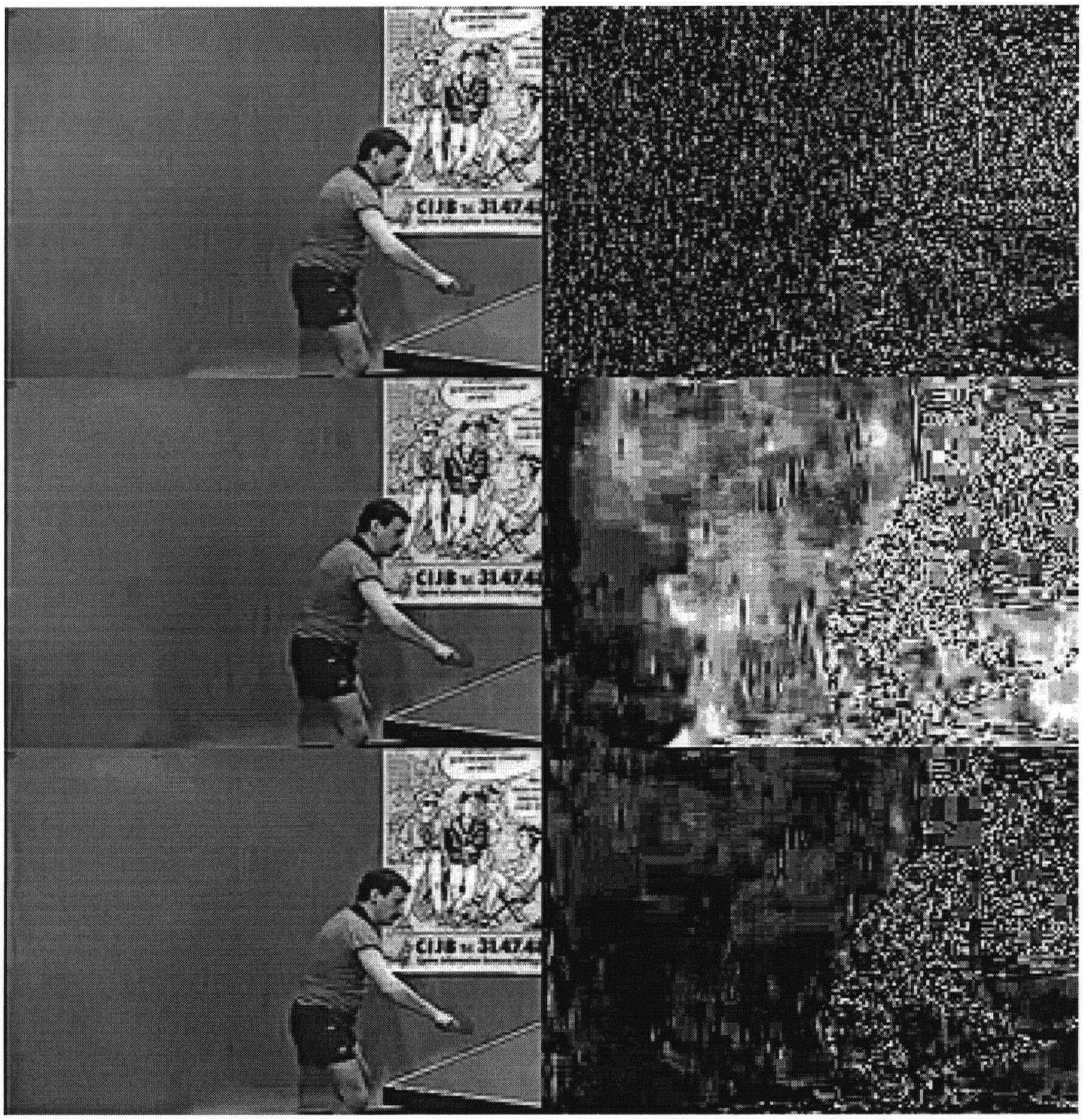


Figure 1.9 Video frames (L) and Frame residual (R)

Thus, bit-rate control in H.264 (and similar codec) is conducted by controlling the Quantization Parameter (QP) and adjusting λ accordingly, using **(1.16)**.

In Figure 1.9, the top row (L) shows an unwatermarked video frame from the Tennis video sequence, compressed using the H.264 codec with a QP=28. On the right, the luma histogram is shown, which illustrates the bit allocation to different regions of the image. It can be seen that more bits are allocated to regions of fine detail. The middle row (L) shows the same frame watermarked using traditional SCS. It can be seem from the corresponding histogram that the bit allocation is almost uniform throughout the image. Finally, the bottom row (L) shows the same frame watermarked using Improved SCS. The histogram on the right shows that bits are allocated in proportion to the encoder's bit-allocation scheme.

Collusion resistance: The basic rule for collusion-resistant video watermarking (Section 1.4) requires that the watermark is strongly adapted to the host signal. This implies that there should be a factor which controls watermark embedding strength, based on *local statistics* of the host signal. SCS does have a watermark scale factor α (Section 1.5.3). But the existing formula for computing the optimum α **(1.15)** has two limitations:

- α is dependent on a single *global statistic*, namely the WNR.
- α has to be *precomputed* for a given WNR.

Due to these reasons, traditional SCS is susceptible to collusion attacks.

2 Improved Scalar Quantization-based Digital Video Watermarking scheme

2.1 Watermark Encoder

In this chapter, we present our proposed watermarking scheme in detail. As explained in Section 1.5.4, traditional SCS has certain limitations if used for video watermarking. We show how our scheme overcomes these limitations. Our scheme builds on the basic ideas of SCS. It is designed specifically for video and is extremely robust to a wide range of attacks. We use a locally adaptive watermark embedding and optimum Rate-Distortion. A unique perceptual mask controls the levels of spatial and temporal distortion, while a built-in bit-rate control ensures optimum watermark bit allocation.

2.1.1 Spread Transform Coding

In our proposed scheme, we use Spread Transform (ST) coding which is a special embedding technique that yields low bit-error rates ([13], [14], [15]). ST combined with SCS is called ST-SCS. We now describe ST coding and how it is adapted to our scheme. Figure 2.1 shows the reconstruction points of two quantizers for embedding one bit in a sample of the host signal. To embed a '0' bit, the host signal is quantized to the nearest 'O' point and to embed a '1' bit, it is quantized to the nearest 'X' point.

Figure 2.2 shows the case of Spread Transform, i.e., a unitary transform has been first applied to the host signal before embedding a bit. The process of applying a unitary transform can be viewed as projecting the host signal onto a vector **v** whose direction is as shown in Figure 2.2. In this case, to embed a '0' bit, the host signal is quantized to the

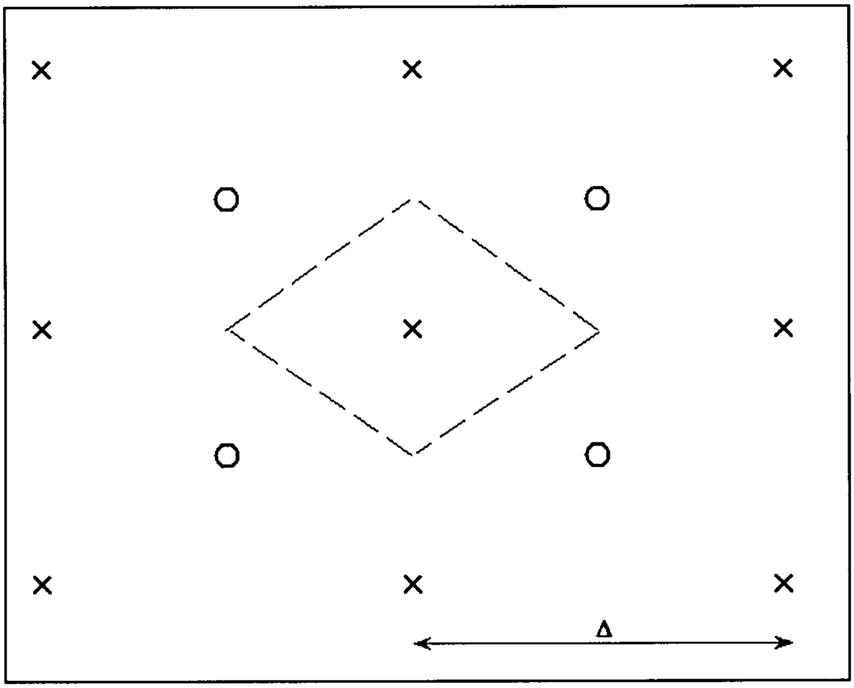


Figure 2.1 Reconstruction points for uniform scalar quantization

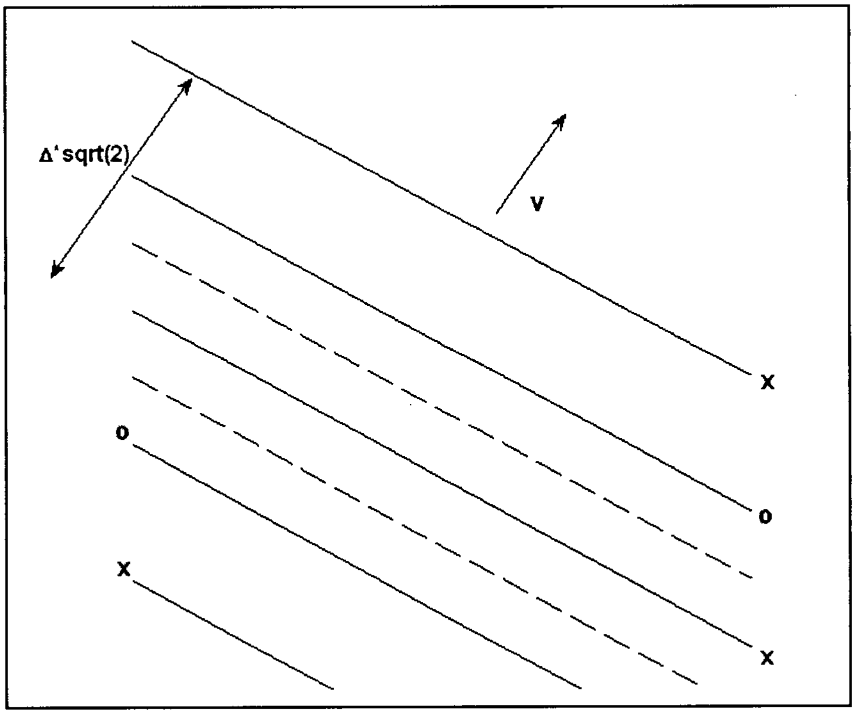


Figure 2.2 Reconstruction points for Spread Transform coding

nearest *line* which is marked with an 'O' and to embed a '1' bit, it is quantized to the nearest *line* marked with an 'X'. It can be observed that in both cases, the minimum distance between adjacent reconstruction points (i.e., an 'O' and an 'X' point or line) is Δ/√2. Thus, the robustness to perturbation due to noise or attacks is the same in both cases. However, the important difference lies in the fact that the *number* of perturbation vectors that can cause decoding errors is higher for Figure 2.1 than for Figure 2.2.

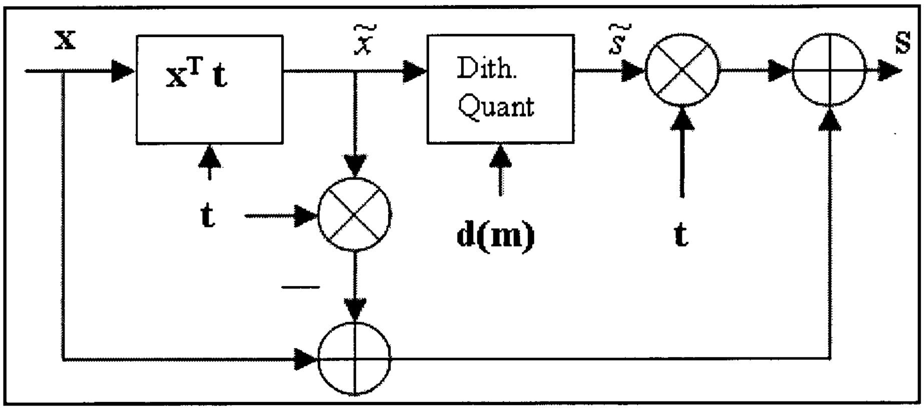


Figure 2.3 Spread Transform Watermarking

Lately, Spread Transform coding [5] has been used in combination with traditional SCS (known as ST-SCS) to improve the bit-error rate of watermarking. We have developed a new watermarking method which borrows ideas from Spread Transform coding and SCS and is specifically designed for video.

2.1.2 Watson's Perceptual Model

In traditional Spread Transform Scalar Costa Scheme (ST-SCS) watermarking, the cover work **x** is projected onto a pseudo-random vector. The disadvantage of this approach is that it does not account for perceptual masking effects of the Human Visual System (HVS). In our scheme, we propose to use a unique perceptual mask sequence **t**, which is derived from the host signal **x** itself, in order to achieve imperceptibility. The generation of **t** differs depending on which type of macroblock is used.

Our perceptual mask is based on Watson's perceptual model [1]. This model estimates the perceptibility of changes in the coefficients of the block-based DCT of an image. This is obtained by first dividing the image into independent blocks of a fixed size (generally 8x8 pixels). If we denote the video frame by **f**, then the i, j^{th} pixel in block number *k* is denoted by

$$\mathbf{f}[i, j, k], 0 \leq i, j \leq 7.$$

For every single block, the resulting DCT matrix is denoted by

$$\mathbf{F}[i ,j, k], 0 \leq \mathrm{i}, \mathrm{j} \leq 7.$$

F[*0, 0, k*] is used to denote the DC component, which represents the average brightness level of that block. Watson's model was originally intended for use in JPEG compression, in order to estimate perceptibility of the quantization noise for each DCT coefficient. Watson's model consists of a frequency sensitivity function, luminance and contrast masking components.

Frequency sensitivity: Watson's model defines a sensitivity table, with each table entry representing the smallest magnitude of the corresponding DCT coefficient in a block that can by perceived by the eye. This magnitude is commonly referred to as the Just Noticeable Difference (JND) amount. A smaller value in the table indicates that the human eye is more sensitive to a change in that particular frequency and hence it can only be changed by a small amount before being noticed. The sensitivity table is a function of several parameters, which include the resolution of the image, block size of the transform and the normal viewing distance of the observer. For normal viewing conditions, the frequency sensitivity function is derived in [1]. This table is shown in Table 2.1. Each table entry is denoted by $\mathbf{fs}[i, j]$.

Table 2.1 WATSON'S DCT FREQUENCY SENSITIVITY TABLE

	0	1	2	3	4	5	6	7 (i)
0	1.404	1.011	1.169	1.664	2.408	3.433	4.796	6.563
1	1.011	1.452	1.323	1.529	2.006	2.716	3.679	4.939
2	1.169	1.323	2.241	2.594	2.988	3.649	4.604	5.883
3	1.664	1.529	2.594	3.773	4.559	5.305	6.281	7.600
4	2.408	2.006	2.988	4.559	6.152	7.463	8.713	10.175
5	3.433	2.716	3.649	5.305	7.463	9.625	11.588	13.519
6	4.796	3.679	4.604	6.281	8.713	11.588	14.500	17.294
7	6.563	4.939	5.883	7.600	10.175	13.519	17.294	21.156
(j)								

Luminance masking: This factor accounts for the effect of the DC-component (i.e. the average brightness of the block) on the frequency sensitivity table. A DCT coefficient can be changes by an amount larger than that indicated in Table 2.1, that block has a higher DC component. Therefore, the frequency sensitivity table $\mathbf{fs}[i, j]$ needs to be adjusted using the DC term for that block. This adjustment factor is given by,

$$\mathbf{fs}_{\mathrm{L}}[i, j] = \mathbf{fs}[i, j] * (\mathbf{F}[0, 0, k] \,/\, \mathbf{F}_{0,0})^{0.649} \quad , \qquad (2.1)$$

where $\mathbf{fs}_{\mathrm{L}}[i, j]$ is the luminance masked threshold and $\mathbf{F}_{0,0}$ is the average of all DC coefficients of the image. $\mathbf{F}_{0,0}$ can also be set to the expected brightness of the video frame. We choose $\mathbf{F}_{0,0} = 128$ in our experiments, instead of calculating it separately for each frame.

Contrast masking: This factor takes into account the effect of visibility of a change in one frequency due to the energy present in that particular frequency. A contrast masking threshold $\mathbf{fs_C}[i, j]$ is generated from the luminance masked threshold as shown below:

$$\mathbf{fs_C}[i, j, k] = \max \{ \mathbf{fs_L}[i, j, k] , | \mathbf{F}[i ,j, k] |^{0.7} * (\mathbf{fs_L}[i, j, k])^{0.3} \}. \qquad (2.2)$$

$\mathbf{fs_C}[i, j, k]$ represents the thresholds or slacks for the individual DCT coefficients. These slacks represent the amounts by which the individual coefficients maybe changes, before resulting in a perceptible change in the block (i.e. 1 JND).

2.1.3 Generation of the Unique Perceptual Mask Sequence

Intra macroblocks: In video coding, Intra macroblocks are those which are coded without reference to any other macroblock. In other words, Intra macroblocks use spatial redundancy of the image in order to achieve compression. Therefore, we consider spatial masking effects are considered. The procedure for obtaining the perceptual mask is detailed below:

1. First, a 3x3 Gaussian low-pass filter, with zero mean and variance = 0.5, is applied to the macroblock in order to mitigate the effect of noise. This filter configuration was chosen because it was found to be the best compromise between speed and performance.
2. Then, for each given macroblock, the transform coefficients are obtained. The exact transform depends on the video coding standard used. For example, MPEG-2 uses a real valued DCT transform, whereas H.264 uses an integer-based transform, which is a close approximation to the actual DCT. Let $\mathbf{f_m}[i, j]$ denote

the spatial macroblock values. The actual block size used depends again on the video coding standard. Some typical block sizes are 8x8 for MPEG-2 and 16x16 and 4x4 for H.264 ([16], [17]).

$$\mathbf{F}_{\mathbf{m}}[i, j] = \mathrm{T}\ \{\ \mathbf{f}_{\mathbf{m}}[i, j]\ \} \quad , \qquad (2.3)$$

where T denotes the transform used.

3. Next, the image-independent frequency sensitivity value $\mathbf{fs}[i, j]$ values from Table 2.1 are selected and used in (2.1) to obtain the image-dependent luminance masked thresholds:

$$\mathbf{fs}_{\mathrm{L}}[i, j] = \mathbf{fs}[i, j] * (\mathbf{F}_{\mathbf{m}}\ [0, 0]\ /\ \mathbf{128}\ )^{0.649}. \qquad (2.4),$$

4. Then, the contrast masked thresholds are obtained by applying (2.2) to the luminance mask thresholds:

$$\mathbf{p} = \mathbf{fs}_{\mathrm{C}}[i, j] = \max\ \{\ \mathbf{fs}_{\mathrm{L}}[i, j]\ ,\ |\ \mathbf{F}_{\mathbf{m}}\ [i, j]\ |^{0.7}\ *\ (\mathbf{fs}_{\mathrm{L}}[i, j]\ )^{0.3}\ \} \quad , \ (2.5)$$

where p represents the slacks for this macroblock.

5. For Intra macroblocks the final perceptual mask sequence **t** is given by:

$$\mathbf{t} = \mathbf{p}\ /\ |\mathbf{p}| \ , \qquad (2.6)$$

where $|\mathrm{p}|$ denotes the magnitude of the vector p. This final step ensures that the final perceptual mask sequence values are normalized, i.e. in the range 0-1. This is necessary to preserve the basic property of Spread Transform watermarking as laid out in [14] and [15].

Inter macroblocks: These macroblocks are coded using a previously coded macroblock as a reference. The reference macroblock may be another Intra or Inter macroblock.

Motion estimation is first used to find the best match macroblock in the reference frame. Next, motion compensation is performed – the best match macroblock is subtracted from the current macroblock. The difference signal, or motion residual, is then coded along with the motion information. The motion information is called the motion vector and consists of horizontal and vertical pixel displacement values $\mathbf{m_x}$ and $\mathbf{m_y}$. Therefore, both spatial and temporal masking must be considered.

Previous research has shown that watermark artifacts, such as "mosquito" effects and flicker, are visible in the fast moving regions of a frame [18]. These artifacts correspond to regions with a large motion vector values. For this reason, the strength of the watermark should be reduced in such regions. This is achieved by weighting the perceptual mask by the inverse of the motion vector magnitude. Thus, for Inter macroblocks, the perceptual mask is first computed using (2.3) - (2.6). Then, the motion vector magnitude $|\mathrm{mv}|$ is computed using

$$|\mathbf{mv}| = \sqrt{(\mathbf{m_x}^2 + \mathbf{m_y}^2)}\,. \qquad \mathbf{(2.7)}$$

Then, the final perceptual mask sequence is given by:

$$\mathbf{t} = \mathbf{p} \,/\, |\mathbf{mv}|\ . \qquad (2.8)$$

2.1.4 Watermark Embedding using Improved ST-SCS

Once the perceptual mask $\mathbf{t}$ is generated for the current macroblock, the projection of $\mathbf{x}$ onto $\mathbf{t}$ is found. This operation yields a scalar quantity:

$$\tilde{x} = \mathbf{x}^{\mathrm{T}}\mathbf{t}.$$

In our scheme, $\mathbf{x}$ represents the *transform domain* coefficients of Intra and Inter coded macroblocks. The watermark key K is used to generate the random scalar value $k \in [0,1)$. For binary ST-SCS, the equation for embedding a '0' bit is obtained by putting d=0 and D=2 in **(1.10)**

$$\tilde{s} = Q_{\Delta}\left\{\tilde{x} - \Delta k\right\} - \left(\tilde{x} - \Delta k\right) . \qquad \textbf{(2.9)}$$

Similarly, embedding a '1' bit is possible by setting d=1 and D=2 in **(1.10)**:

$$\tilde{s} = Q_{\Delta}\left\{\tilde{x} - \Delta\left(0.5 + k\right)\right\} - \left(\tilde{x} - \Delta(0.5 + k)\right) . \qquad \textbf{(2.10)}$$

The components of $\mathbf{x}$ that are orthogonal to $\mathbf{t}$ are equal to $\mathbf{x} - \tilde{x}\mathbf{t}$. These components are not altered during the embedding process and are for this reason they are added back to the watermark data. Therefore, the final watermarked data s is obtained by combining (4) with the orthogonal components:

$$\mathbf{s} = (\tilde{\mathbf{x}} + \alpha\tilde{\mathbf{s}})\mathbf{t} + (\mathbf{x} - \tilde{\mathbf{x}}\mathbf{t}) . \qquad \textbf{(2.11)}$$

2.1.5 Selection of the Watermark Scale Factor α using Rate-Distortion Optimization

Traditional ST-SCS uses a fixed α that is pre-computed from global statistics **(1.15)**. In contrast, our method uses a locally adaptive value for α which is computed in real-time from a combination of local and global statistics. As a result, we obtain stronger control over the watermark scale factor, which makes our watermark to adapt better to the host signal characteristics. This makes our method much more robust than traditional ST-SCS.

During video encoding, several coding parameters such as macroblock prediction modes, motion vectors and transform coefficient quantization levels have to be determined [19]. Since natural video has widely varying spatial and temporal (motion) content, the selection of different coding options for different parts of the image becomes necessary. Therefore, the task of the video coder is to find a set of coding parameters so that a trade-off between the video bit-rate and distortion (R-D) is achieved. This means that for a given video bit-rate, the encoder has to find the combination of coding options that minimizes the distortion.

Optimization using Lagrangian Techniques: Lagrangian bit-allocation techniques for R-D coding have been widely accepted in recent video codec development, due to their effectiveness and simplicity. Adding a watermark to a video stream may also affect the bit rate and quality of the image. It is, therefore, highly desirable that the watermark embedding procedure incorporates R-D optimized coding in order to compute the optimum watermark for different regions of a video frame. The general Lagrangian technique is described now [20]. Consider N source samples that are to be coded using R-D optimization. Let the samples be

$$S = \{S_1, S_2 .... S_N\} \quad . \qquad \textbf{(2.12)}$$

Now, each source sample can be coded using several possible coding options represented by an index out of the set O_n:

$$O_n = \{ O_{n1}, O_{n2} .... O_{nK} \} \quad . \qquad \textbf{(2.13)}$$

Let $I_n \in O_n$ be the selected index to code the source sample S_n. Then, the coding options which are need to code S are given by the components

$$I = \{ I_1, I_2 .... I_N \} \quad . \qquad \textbf{(2.14)}$$

Now, the problem of finding the correct combination of coding options that minimizes the distortion for the given set of source samples S, subject to a rate constraint R_c can be written as:

$$\min_I D(S,I) \ , \qquad \textbf{(2.15)}$$

$$\text{subject to } R(S,I) \leq R_c.$$

Here D(S,I) represents the total distortion which results from the quantization of S with the particular combination of coding options I. R(S,I) denotes the bit-rate which results after quantization of S. Now, (2.15) represents a constrained formulation. However, in practice, an equivalent unconstrained formulation is employed, namely

$$I = \text{argmin}_I \ J(S,I|\lambda), \qquad (2.16)$$

$$\text{with } J(S,I|\lambda) = D(S,I) + \lambda^*R(S,I).$$

and $\lambda \geq 0$ being the Lagrangian parameter. Equation (2.16) represents the unconstrained solution to a discrete optimization problem. The solution obtained in (2.16) is optimal because if a rate constraint R_c corresponds to λ, then the total distortion D(S,I) is minimum for all combinations of coding options which result in a bit-rate less than or equal to R_c. Assuming that the distortion and rate measures are additive in nature, and that these two quantities are only dependent on the choice of the coding options for each source sample, the simplified Lagrangian cost function can be written as .

$$J(S_n,I|\lambda) = J(S_n,I_n|\lambda), \qquad (2.17)$$

which can be reduced to

$$\min_{I} \sum_{n=1}^{N} J(S_n, I|\lambda) = \sum_{n=1}^{N} \min_{I_n} J(S_n, I_n|\lambda). \qquad (2.18)$$

Now, it is very easy to solve (2.18) by independently selecting the coding option for each $S_n \in S$. This particular formulation was first suggested by Shoham and Gersho in [21].

Lagrangian technique for selecting α : We use the Lagrangian multiplier technique to compute the locally optimum value of α at the macroblock level. The Lagrangian technique used for video coding is easily extended to work with our proposed watermarking scheme, because in both cases the distortion is caused due to scalar quantization of the source samples. The simplified Lagrangian cost function for a particular value of α is:

$$J_{\alpha} = D_{\alpha} + \lambda_{w} E_{\alpha}, \qquad \textbf{(2.19)}$$

where D_{α} is the distortion (sum of squared differences or SSD) between the cover work **x** and the watermarked work **s**, λ_w is the Lagrangian parameter for watermark embedding and is dependent on the choice of the video standard used for encoding ([20], [22]), and E_{α} is the decoding error = $||D_d| - |D_e||$. We define D_d as the *decoded distance* and D_e as the *expected distance.* D_e is equal to 0 if the embedded message bit d=0, and it is equal to $\pm\Delta/2$ if d=1. To obtain D_d, the watermarked data **s** is projected onto the perceptual mask **t,** which results in the scalar $\tilde{e}$. The quantization of $\tilde{e}$ yields D_d:

$$D_d = Q_{\Delta}\{\tilde{e} - k\ \Delta\} - (\tilde{e} - k\ \Delta). \qquad \textbf{(2.20)}$$

For each macroblock, we compute the value of α which minimizes the Lagrangian cost function **(2.19)**. This value of α is the Rate-Distortion optimum watermark scale factor which we use in our method.

2.1.6 Watermark Bit-rate Control

Watermarked video generally requires many more bits than unwatermarked video, especially at low video bit-rates. Therefore, it is desirable to have a bit-rate control scheme in order to find the optimum trade-off between the fidelity of the watermark and that of the host signal. This scheme should determine the best allocation of available watermark bits between different watermarked macroblocks.

In video coding, the overall video bit-rate is determined by three factors:

- Prediction-mode decisions
- Motion vector choices
- Displaced Frame Difference (DFD) or residual coding fidelity

Of these, the most important factor for controlling the bit-rate is the residual signal coding fidelity, which is controlled by choosing a suitable quantization step-size for the transform coefficients. A larger step size results in a lower bit-rate, but also a larger amount of distortion. Therefore, the choice of optimal step-size for quantization is related to the choice of the relative emphasis given to rate and distortion in **(2.16).** While the choice of the quantization step-size must be communicated to the decoder, λ is an encoder design issue and is not needed by the decoder. The bit-rate can either be controlled to maintain a local average bit-rate over a period of time, or it can be allowed to vary depending on the scene content.

When embedding a watermark using our proposed scheme, we have to deal with similar issues as in general video bit-rate control. Since our watermark is quantization-based, a lower quantization step results in a higher signal fidelity, but also causes many more bit errors during watermark decoding. Therefore, controlling the watermark bit-rate according to local scene content is important to ensure the best trade-off between fidelity and bit-errors. Our scheme is designed in such a way that it achieves watermark bit-rate control simply by changing the quantization step Δ which is used to embed the watermark in **(2.9)**, **(2.10)** and **(2.20)**. Therefore, our scheme has a built-in mechanism for watermark bit-rate control, through the parameters Δ and λ_w. This is an important advantage over existing schemes, such as [18], which require an explicit bit-rate controller. In Chapter 3, we explain how bit-rate control is achieved when we implement our scheme on H.264/AVC.

2.2 Watermark decoder

An important advantage of our method is that the watermark can be decoded from the partially decompressed video bit stream, since the watermark is embedded in the transform coefficients. Decoding of the watermark requires knowledge of the secure key K, which is needed to generate the pseudorandom scalar k. The perceptual mask $\mathbf{t'}$ is computed for this macroblock as explained in Section 2.1.3. The reconstructed transform coefficients x' are projected onto $\mathbf{t'}$ to obtain the scalar projection $\tilde{y}$.

$$\tilde{y} = \mathbf{x'}^T\mathbf{t'} \quad . \qquad \textbf{(2.21)}$$

This projection is then quantized using **(2.9)**, with a step size of Δ

$$|y| = Q_{\Delta}\{\tilde{y} - k\ \Delta\} - (\tilde{y} - k\ \Delta)\ . \quad \textbf{(2.22)}$$

As in traditional ST-SCS, we use simple hard decision decoding is to extract the message $\hat{m}$. The extracted bit is '0' if $|y| \leq \Delta/4$ and '1' otherwise.

3 Watermarking of H.264/AVC Video

3.1 H.264/AVC Video Watermarking Challenges

Although H.264/AVC ([16], [17]) is the latest and most advanced video coding standard, to this date there are very few watermarking schemes designed for H.264. On investigation, it becomes clear that there are several challenges for any H.264 watermarking scheme. First, the compression efficiency of H.264 presents a major challenge for any video watermarking approach. One of the main challenges is that in H.264 even the Intra-frames consist mainly of residual data which have very small initial values. This means that after quantization, the majority of the coefficients have zero values. Therefore, adding a watermark without affecting the picture quality or the bit rate is extremely difficult.

Second, H.264 achieves bit rate reduction for the Intra-frames by using spatial prediction for Intra macroblocks, a major departure from previous coding standards like MPEG-2 and MPEG-4. It supports 3 types of Intra coding: Intra_4x4, Intra_16x16 and I_PCM. In Intra_16x16, the entire 16x16 Macroblock is predicted from the 16 top and left neighboring pixels. There are 4 Intra_16x16 modes: Vertical, Horizontal, DC and Plane mode. In Intra_4x4, each 4x4 luma block is separately predicted using the top and left pixels of previously encoded neighbors (Figure 3.1). There are a total of 9 directional Intra_4x4 modes. The I_PCM coding type is used to bypass the prediction and transform steps.

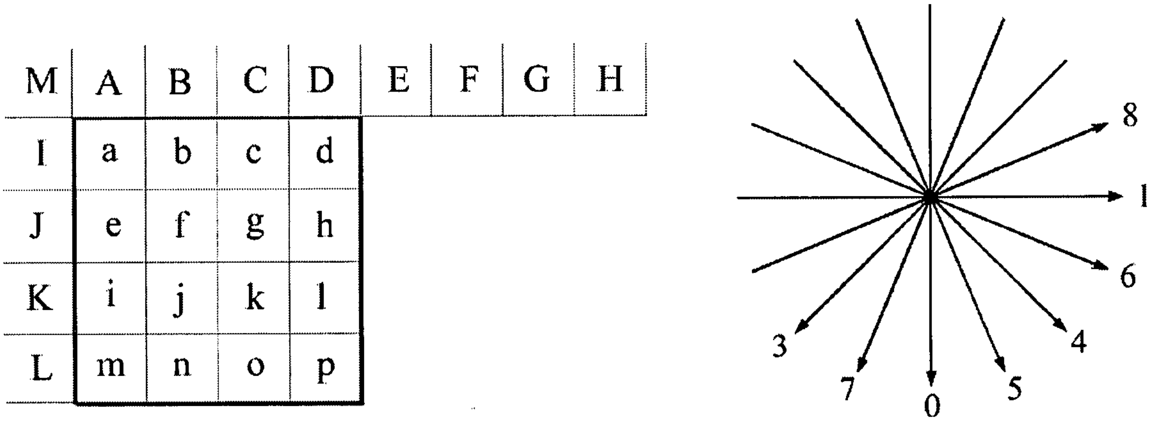
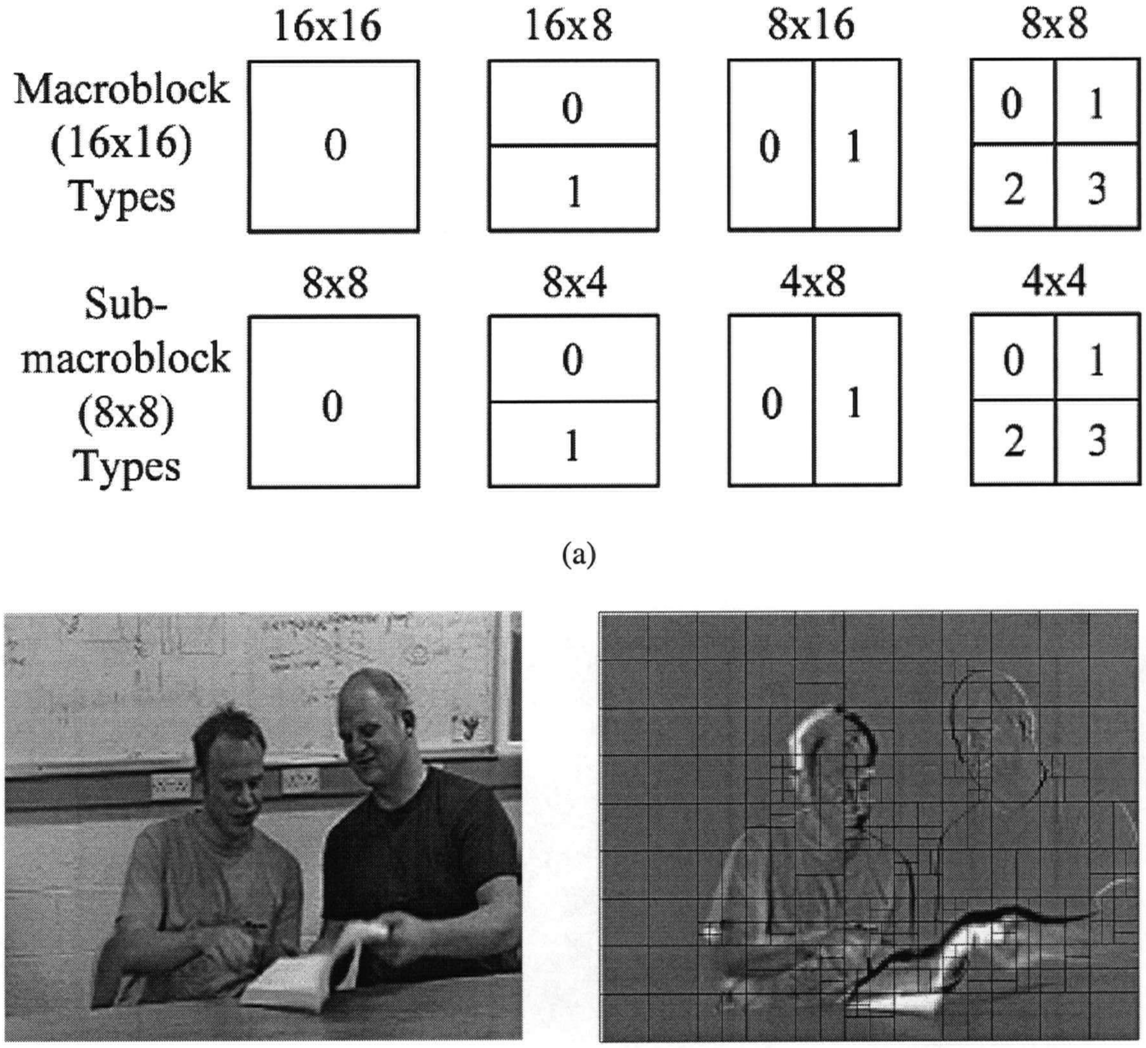


Figure 3.1 Intra_4x4 prediction mode (L) and the 9 possible prediction directions (R)

Third, Inter macroblocks use variable block-size motion compensation. The different sizes include 16x16, 16x8, 8x16 and 8x8. The 8x8 partition can be further divided into 8x4, 4x8 or 4x4 blocks. A motion vector is transmitted for each partition. Sometimes, due to high motion complexity, a macroblock maybe divided into several smaller partitions, each with its own motion vector (Figure 3.2). Therefore, while deriving the perceptual mask sequence t for this macroblock as explained in Section 2.1.3, we divide the Watson's perceptual mask p by the motion vectors for the corresponding regions.

Fourth, for both Intra and Inter macroblocks, Rate-Distortion optimized coding is used to select the best prediction modes [20]. The best prediction is subtracted from the original values to obtain the residual data. The challenge here is that the embedded watermark should not affect the Rate-Distortion optimized coding decision.



(a)

(b)

(c)

Figure 3.2 Inter prediction on H.264; (a) Variable block sizes for motion-compensation; (b) Original frame (c) Residual motion compensated frame showing block sizes used

Another challenge that H.264 poses to watermarking is that it uses an entirely integer transform (Figure 3.3). This presents a major concern for traditional Spread Spectrum watermarking schemes, which embed watermarks drawn from a Gaussian distribution. In summary, H.264 uses 2 types of transforms: an integer transform for the luminance residual data and an additional Hadamard transform for the 4x4 array of the luminance DC coefficients (only in Intra_16x16 mode). In Intra_16x16 mode, much of the energy is concentrated in the DC coefficients of each 4x4 block. The additional transform further concentrates the energy into a smaller number of significant coefficients. After transform, the coefficients undergo scalar uniform quantization, with

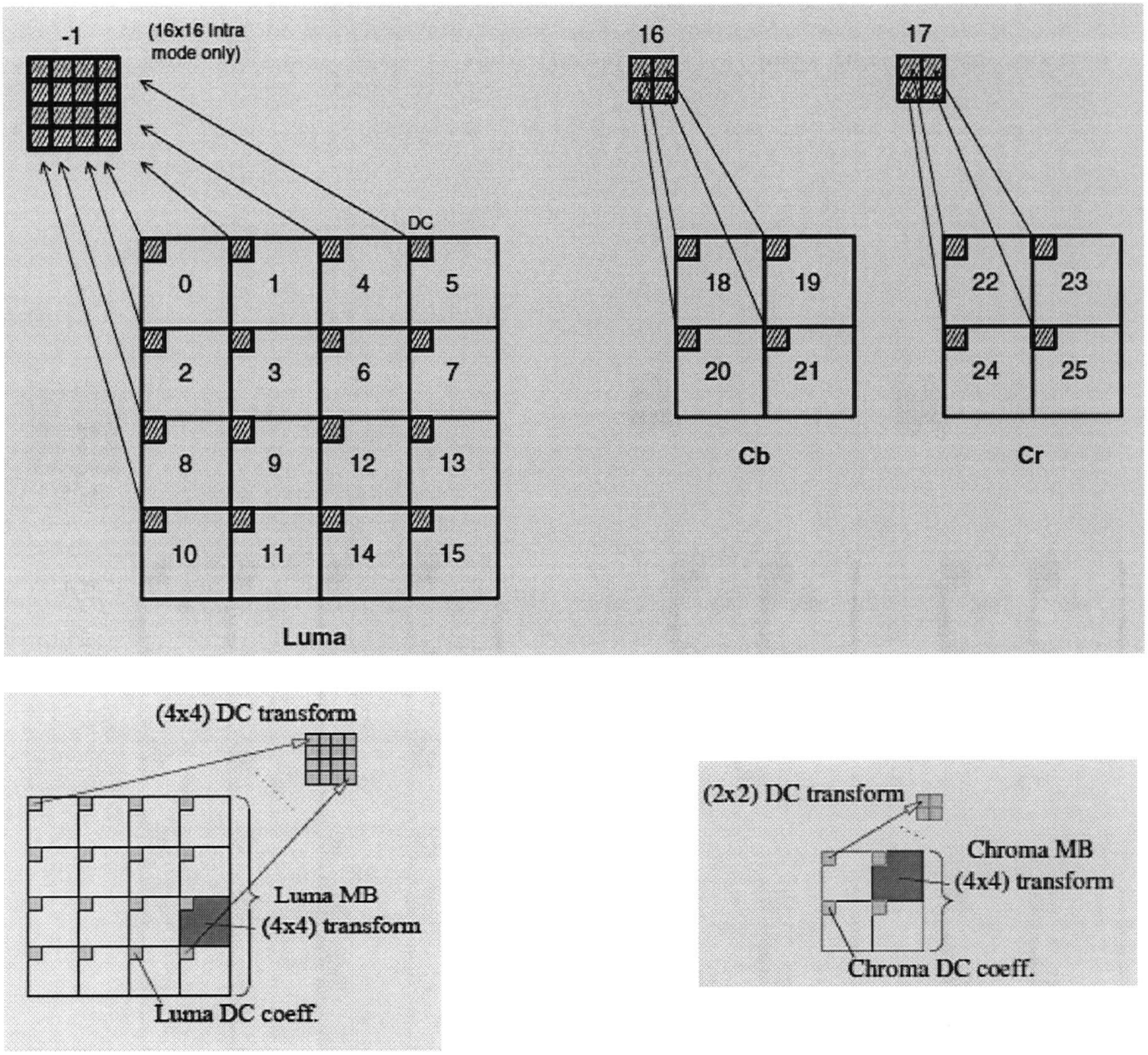


Figure 3.3 Integer transform in H.264/AVC

step size defined by the Quantization Parameter (QP). QP can take values between 0-51, the quantization step (Qstep) doubling for every increase of 6 in QP [23].

3.2 Watermark embedding using the proposed scheme inside the H.264/AVC encoder

The way we address the above challenges and design our watermarking method to work within H.264 is described now. First, we consider watermarking the luminance components of both the Intra and Inter macroblocks. Our scheme operates on the integer transform coefficients of the macroblock residual data. This is possible since we designed our method not to make any assumptions about the nature of the host signal x (please see embedding equations **(2.9)** to **(2.11)**.

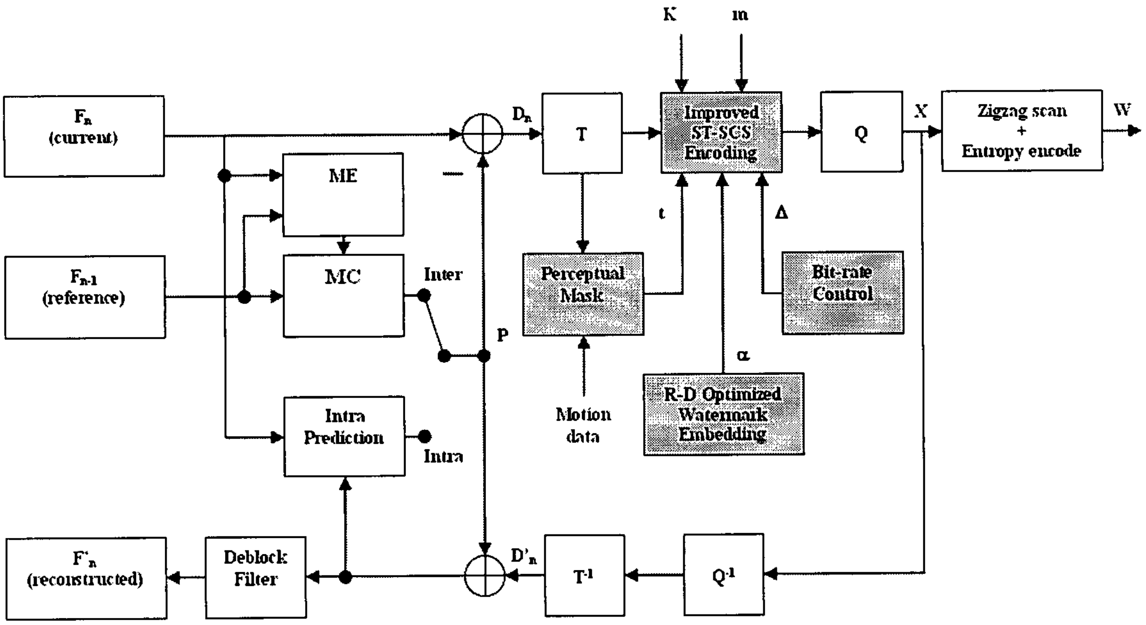


Figure 3.4 Watermark embedding using our proposed scheme inside the H.264 encoder

The entire watermarking embedding process as we implemented inside the existing H.264/AVC encoder is shown in Figure 3.4. It can be observed that our scheme is implemented in-the-loop. The perceptual mask t is first computed from the transform domain data at the macroblock level. Next, the Rate-Distortion optimized local watermark scale factor α is computed, using the Lagrangian optimization technique explained in Section 2.1.5. Our scheme uses a bit-rate control scheme which is related to the H.264 encoder bit-rate control mechanism, thereby eliminating the need for an external bit-rate controller. Using all these blocks, the watermark is embedded into the transform coefficients. The watermarked coefficients are then passed on to the Quantization and Entropy coding blocks of the encoder.

3.2.1 Selection of Transform Coefficients

For macroblocks predicted using the Intra_16x16 or Inter_16x16 modes, the Hadamard coefficients are watermarked. The reason for selecting these coefficients is that in Intra_16x16 mode, the additional transform for the DC coefficients concentrates most of the macroblock energy into a few Hadamard coefficients. For macroblocks predicted using the Intra_4x4 mode as well as the remaining Inter modes (16x8, 8x16, 8x8 and its associated modes), we watermark the integer transform coefficients.

3.2.2 Selection of Scalar Quantizer Step-size Δ

Let $QP_{H.264}$ denote the Quantization Parameter (QP) value used in H.264, Qstep denote the H.264 quantization step size, QP_W denote our watermark quantization parameter and Δ. denote our watermark quantization step-size. The relation between Qstep and $QP_{H.264}$ is as follows:

$$Qstep = 0.6282 * \exp(QP_{H.264}*0.1155),\ 0 \leq QP_{H.264} \leq 51.$$

The behavior of the above equation is such that for values of $QP_{H.264}$ in the range 0-30, the corresponding Qstep changes by very small amounts. However, for $QP_{H.264}$ between 31-51 the Qstep increases very rapidly. Our main objective is to control the distribution of the watermark bits in such a way that minimum Bit-Error Rate (BER) is achieved, independent of the video bit-rate. We achieve this by establishing a relationship between our watermark step size QP_w and $QP_{H.264}$. Evaluations over a large set of video sequences indicated that the minimum BER is obtained when:

$$QP_W = 48,\ 0 \leq QP_{H.264} \leq 30\,, \tag{3.1}$$

and

$$QP_W = 1.329 * QP_{H.264} + 6.768,\ 31 \leq QP_{H.264} \leq 51\ . \tag{3.2}$$

The relationship between QP_w and Δ is exactly the same as that between $QP_{H.264}$ and Q_{step}. Therefore, equivalently,

$$\Delta = 160,\ 0 \leq QP_{H.264} \leq 30\ \text{, and}$$

$$\Delta = 0.6882 * \exp(QP_{H.264}*0.1686),\ 31 \leq QP_{H.264} \leq 51.$$

Thus, there is a close relationship between the equations for Qstep and Δ. Both equations represent exponential curves, with an initial slow ascent between 0-30 and then a rapid increase in the range 31-51. This guarantees that our watermark robustness is constant at all different compression rates.

3.2.3 Watermark Bit-rate Control and Rate-Distortion optimization in H.264/AVC

Our watermark is embedded on the transform coefficients before the Quantization process. The reason for this is that if the watermark was embedded in the non-zero coefficients obtained after quantization, it would have an adverse effect on the bit-rate of the video. In H.264, video bit-rate control is achieved through proper selection of the Quantization Parameter ($QP_{H.264}$), which controls the quantization step-size for the transform coefficients. A larger QP value results in lower bit rates and increased picture distortion. The trade-off between the bit-rate and distortion is determined by the proper choice of $QP_{H.264}$. It has been shown in [19] that there is a strong relationship between the Lagrangian parameter λ used for R-D coding and $QP_{H.264}$:

$$\lambda = 0.85 * 2^{(QP_{H.264} - 12)/3} \quad . \qquad \textbf{(3.3)}$$

Therefore, bit-rate control in H.264 is conducted by controlling $QP_{H.264}$ accordingly adjusting the value of λ used for R-D coding. Similarly, in our method, the watermark bit allocation is controlled by choosing the step size of the scalar uniform quantizer Δ (or equivalently, QP_W) and adjusting the value of λ_w used in **(2.19)**. The Lagrangian parameter λ_w is computed as:

$$\lambda_w = 0.85 * 2^{(QP_w - 12)/3} \quad . \qquad \textbf{(3.4)}$$

We then use this value of λ_w in **(2.19)** to determine the locally optimum watermark scale factor α. By selecting this value of α ,.we ensure that our watermark will not affect the R-D optimized coding decisions of the H.264 encoder. When the H.264 encoder varies

$QP_{H.264}$ in order to achieve the desired overall video bit-rate, QP_W also changes proportionally since it is related to $QP_{H.264}$ through **(3.1)** and **(3.2)**. Therefore, the watermark bits are allocated in proportion to the H.264 encoder's bit-rate control algorithm. This ensures that the overall video bit-rate is not adversely affected.

The entire Bit-rate control and Rate-Distortion optimized watermark embedding procedure is presented in Figure 3.5.

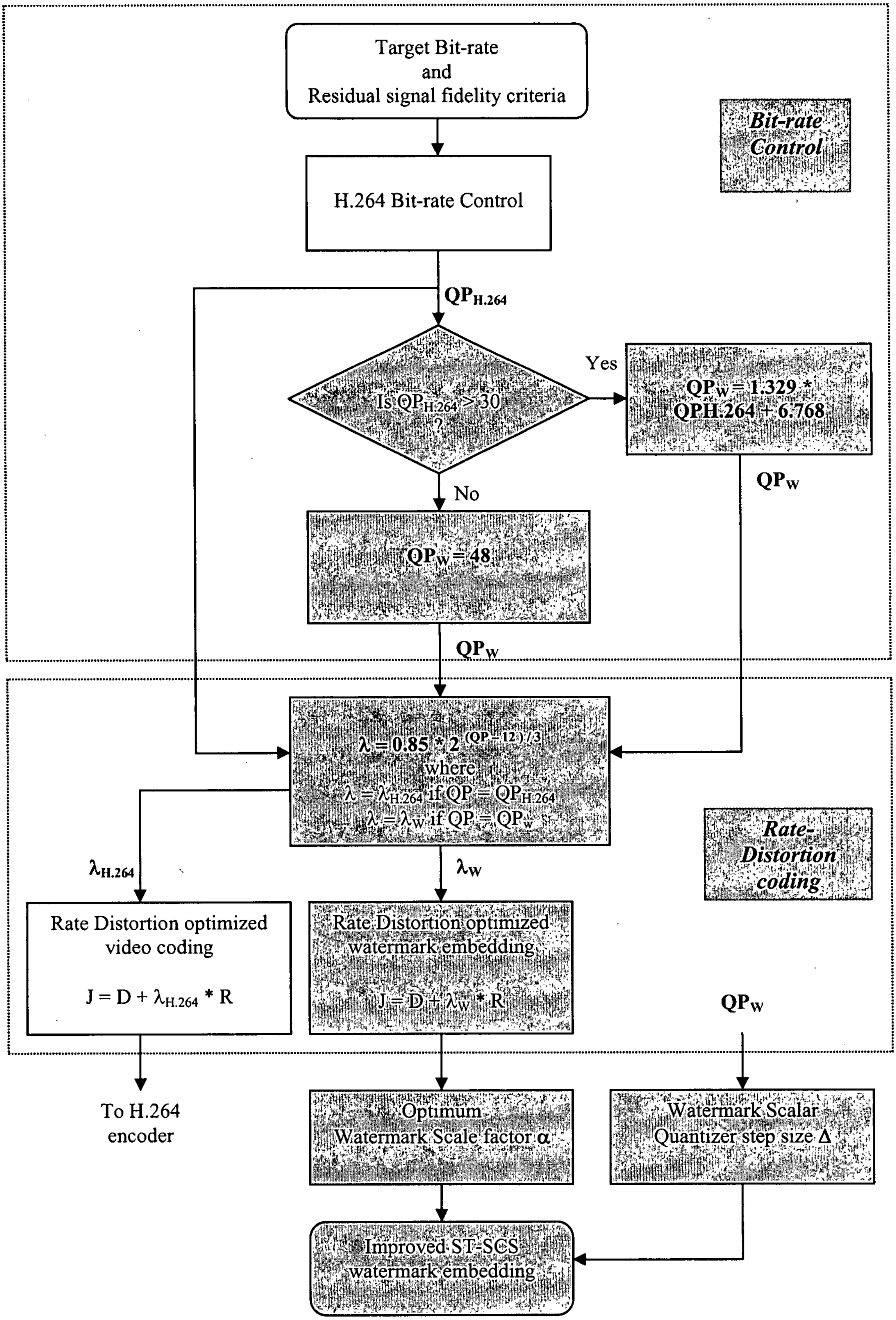

Target Bit-rate
and
Residual signal fidelity criteria
Bit-rate Control
H.264 Bit-rate Control
$QP_{H.264}$
Is $QP_{H.264} > 30$?
Yes
$QP_W = 1.329 * QPH.264 + 6.768$
QP_W
No
$QP_W = 48$
QP_W
$\lambda = 0.85 * 2^{(QP - 12)/3}$
where
$\lambda = \lambda_{H.264}$ if $QP = QP_{H.264}$
$\lambda = \lambda_W$ if $QP = QP_W$
Rate-Distortion coding
$\lambda_{H.264}$
λ_W
Rate Distortion optimized video coding
$J = D + \lambda_{H.264} * R$
Rate Distortion optimized watermark embedding
$J = D + \lambda_W * R$
QP_W
To H.264 encoder
Optimum Watermark Scale factor α
Watermark Scalar Quantizer step size Δ
Improved ST-SCS watermark embedding

Figure 3.5 Bit-rate control and Rate-Distortion coding in our proposed scheme

3.3 *Watermark decoding using the proposed scheme inside the H.264/AVC decoder*

The complete watermark decoding process inside the H.264/AVC decoder is shown in Figure 3.6. The watermark decoding process is relatively simple, compared to the encoding step. First, the perceptual mask **t** is computed from the reconstructed transform coefficients at the decoder. This data is fed to the Improved ST-SCS decoder, which applies **(2.21)** and **(2.22)** and uses simple hard-decision coding to extract the watermark message estimate m'.

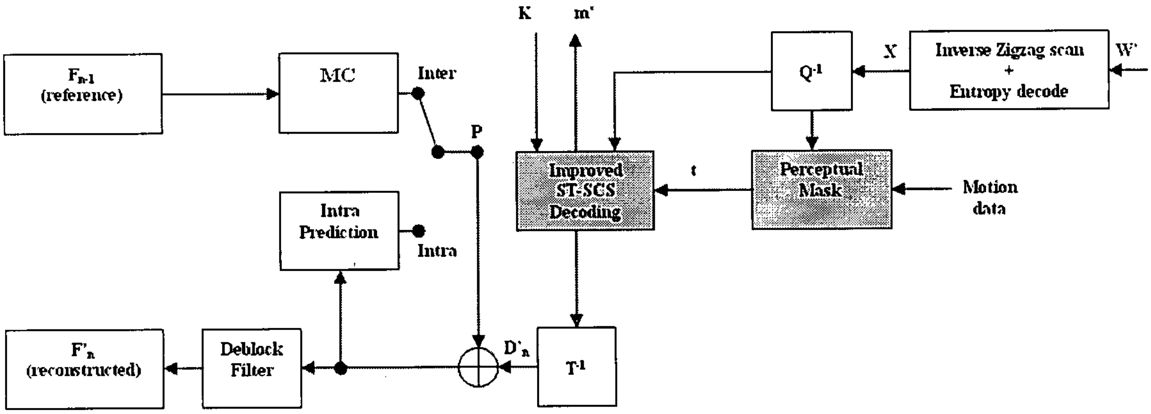


Figure 3.6 Watermark decoding (gray blocks) using our proposed scheme, inside the H.264 decoder (white blocks)

4 Implementation and Experimental results

We used the H.264/AVC reference software version JM9.3 for our implementation [29]. The performance of our scheme was tested on 10 standard video sequences, which represent scenes with varying amounts of camera movement, content motion and spatial detail. The sequences were watermarked and encoded with H.264 at various bit-rates, with a frame rate of 25 frames per second. The Group-of-Pictures (GOP) structure consisted of an Intra (I-) frame followed by 4 Inter (P-) frames. Under the same picture quality, we compared the robustness of our method against the traditional ST-SCS scheme in the following different attack categories:

1. H.264 compression and decompression at different bit-rates from 128 kb/s to 1024 kb/s.

2. H.264 compression and decompression at a fixed bit-rate of 512 kb/s, with different Watermark-to-Noise Ratios (WNR).

3. Transcoding – H.264 bitstreams decompressed and recompressed at the same bit-rate but using a different GOP structure (Intra period = 10)

4. Filtering – 3x3 Gaussian filter of variance 0.5

5. Scaling – spatial scaling with a factor of 75%

6. Rotation – 5°, with bilinear sampling.

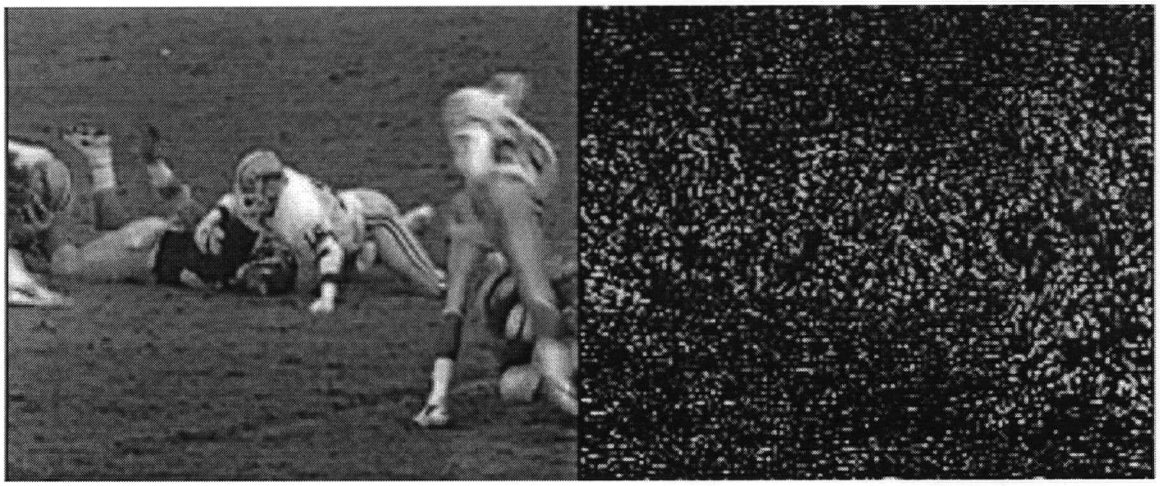
7. Collusion – averaging attack using 5 different watermarked copies of a video sequence at a fixed bit-rate of 512 kb/s.

To ensure the same picture quality we kept the watermark bit-rate (1 bit per macroblock), Spread Transform size (256 elements) and PSNR constant.

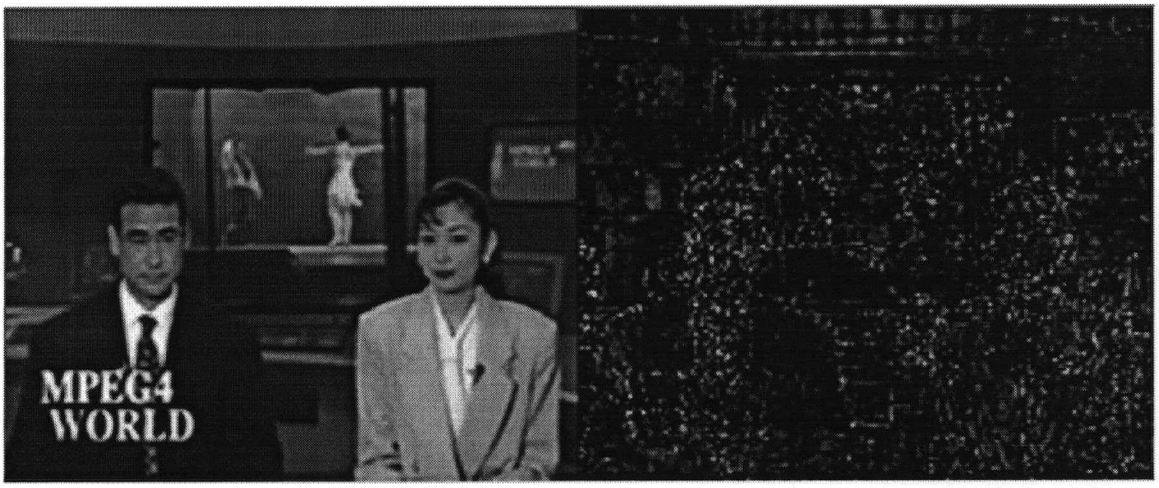
Table 4.1 DESCRIPTION OF VIDEO SEQUENCES USED FOR IMPLEMENTATION

Sequence	Camera motion	Content motion	Spatial detail
Carphone	Still	Moderate / Smooth	Moderate / High
Coastguard	Slow, pan	Moderate / Smooth	Moderate / Low
Football	Still	Fast	High
Foreman	Fast, pan	Moderate / Low	Moderate / High
Flower Garden	Slow, pan	Nil	High
Mother Daughter	Still	Low	Moderate / Low
News	Still	Low	Moderate
Paris	Still	Moderate	High
Tempete	Slow, zoom	High / Random	High
Tennis	Fast, zoom	Fast	Moderate

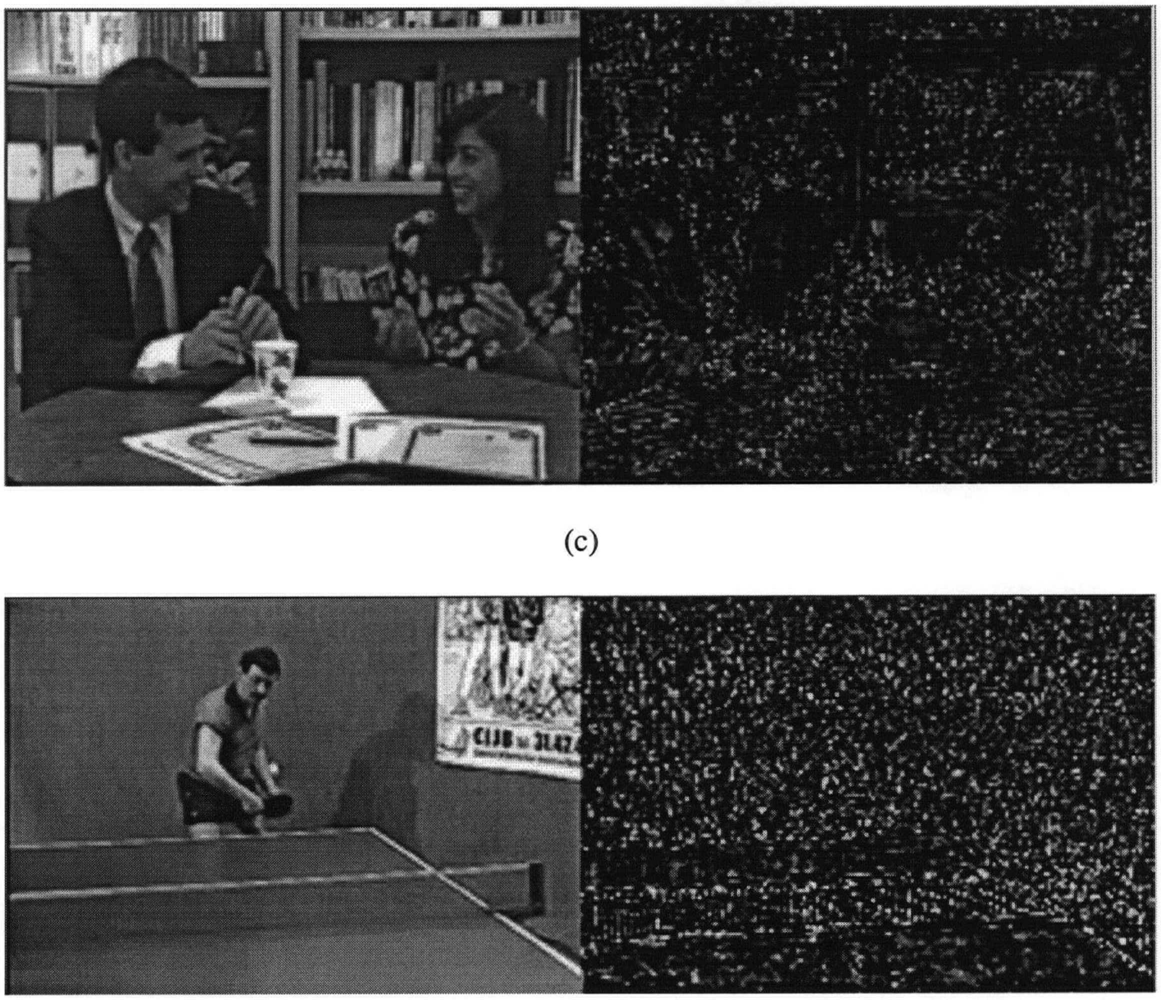
As we observe in Fig. 4.1(a)-(d), which show four representative frames, the resulting watermarked video (obtained in this case by our proposed scheme) maintains excellent subjective quality. The corresponding watermark data that were embedded in the two frames are shown on the right hand. It can be seen that the watermark signal is distributed (as expected) in accordance with the perceptual importance of the different regions of the frame.



(a)



(b)



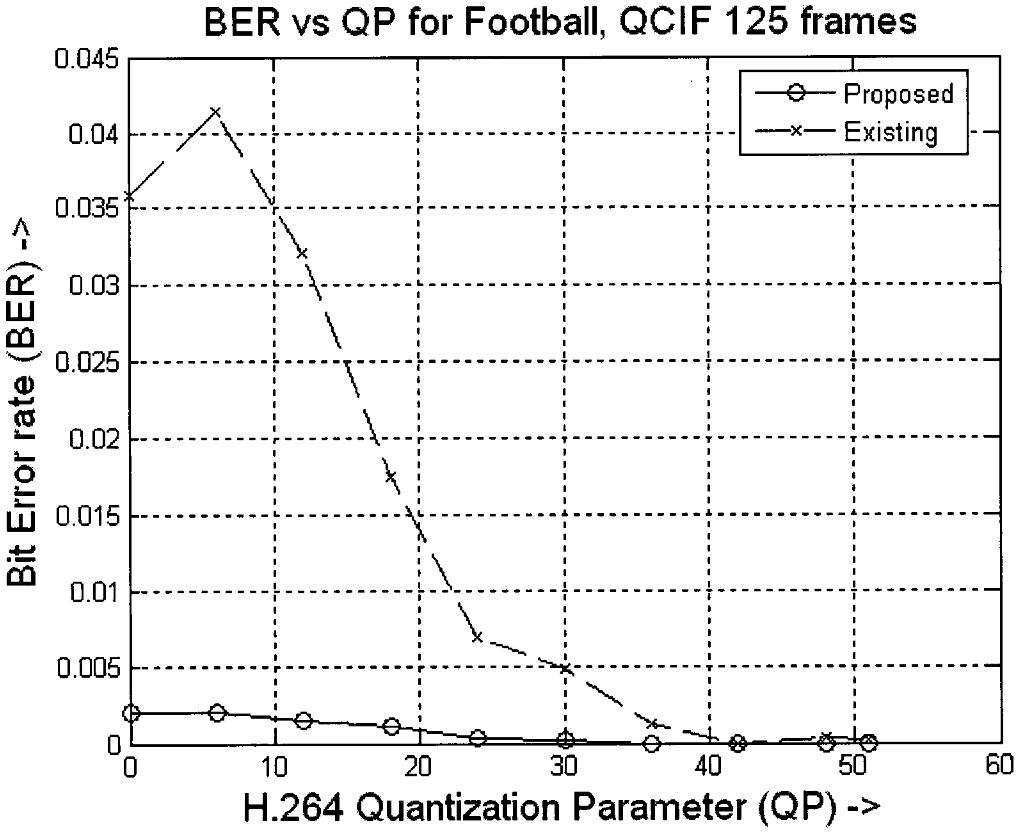
(c)

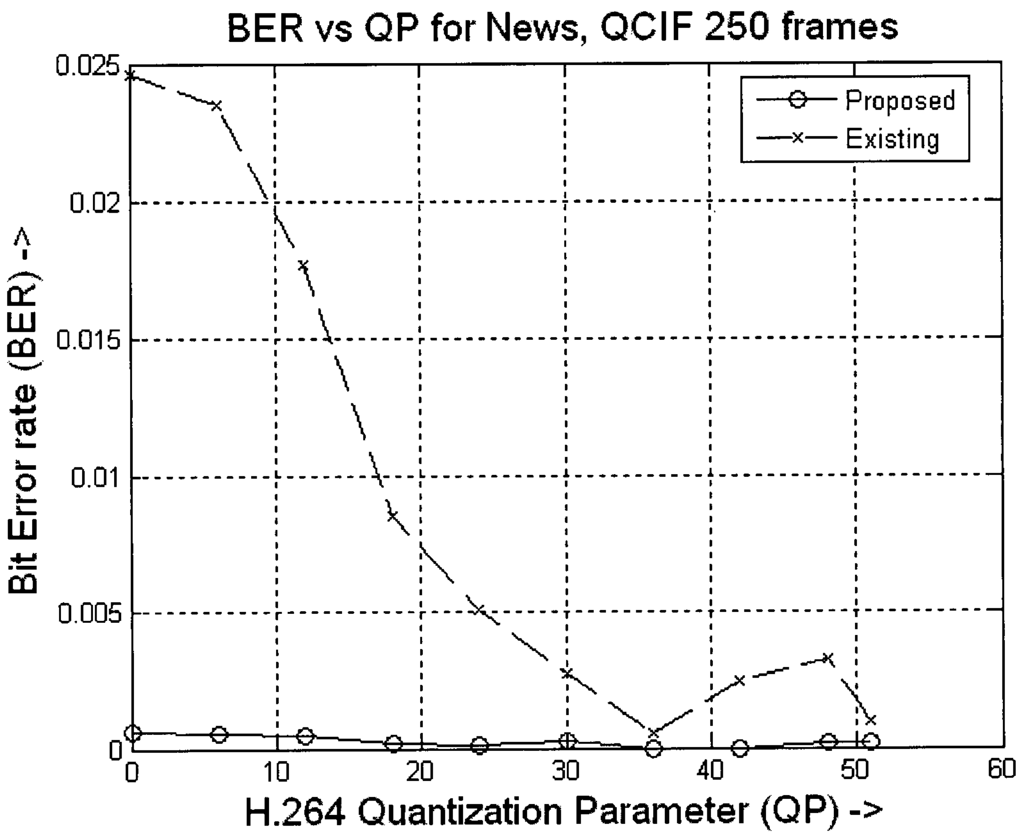
(d)

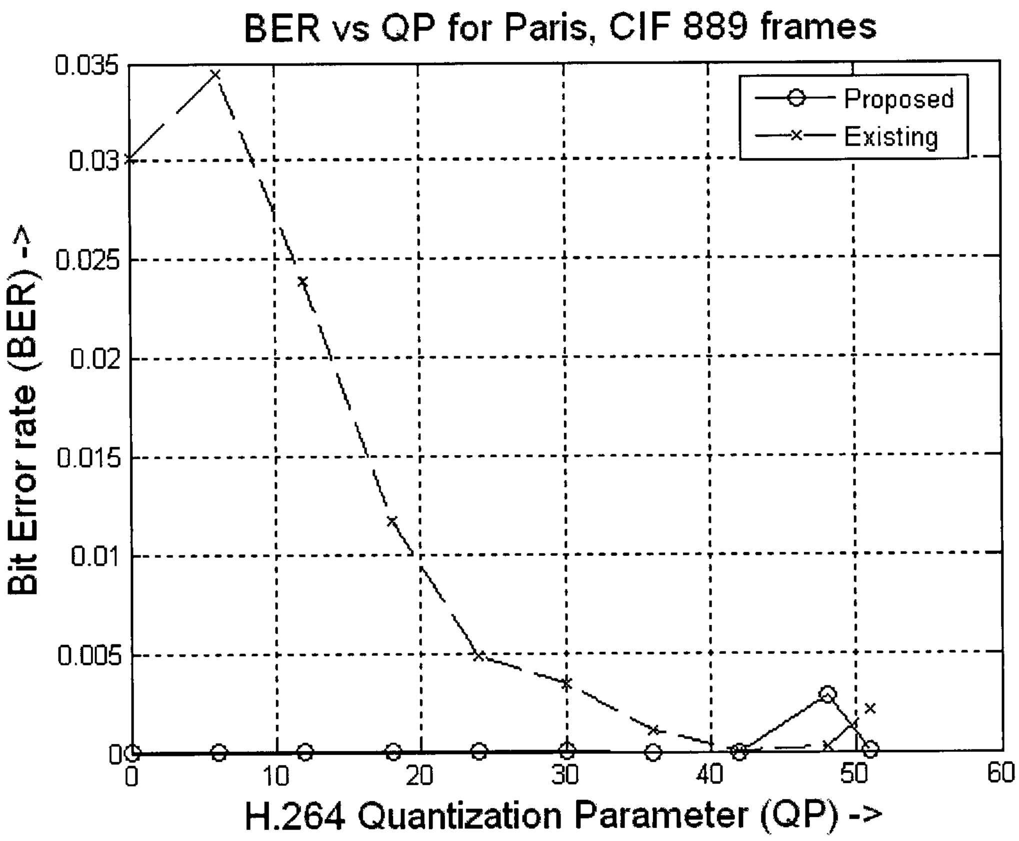
Figure 4.1 Four representative watermarked sequences (a) Football (b) News (c) Paris and (d) Tennis

4.1 H.264 compression and decompression at different bit-rates

The Bit Error Rates (BER) caused by H.264 compression and decompression at various bit-rates, for 4 representative streams are plotted on the following pages. It can be observed that our scheme shows a substantial improvement over traditional ST-SCS. The performance of both watermarking schemes generally decreases with an increase in the video bit-rate. This is because as the bit-rate increases, the H.264 bit-rate controller lowers the quantization step $QP_{H.264}$, resulting in a very high picture quality. As a result, QP_W also decreases proportionally due to (13) and (14). This increases the probability of bit errors during minimum distance decoding, since the distance between the decision levels for the '0' and '1' bits (i.e. 0 and $\tilde{\Delta}/2$) are very close to each other. In case of "Football", our scheme achieves a BER which is 3 orders of magnitude less than that obtained by traditional ST-SCS. Moreover, this improvement is consistent for the various video bit-rates, which indicates that our scheme performs very well in regions of high spatial, as is the case in "Football". On an average, our scheme achieved BERs of about 2 orders of magnitude less than ST-SCS for the 10 sequences, at the same picture quality.




BER vs QP for News, QCIF 250 frames
Proposed
Existing
Bit Error rate (BER) ->
0.025
0.02
0.015
0.01
0.005
0
0
10
20
30
40
50
60
H.264 Quantization Parameter (QP) ->


BER vs QP for Paris, CIF 889 frames
Proposed
Existing
Bit Error rate (BER) ->
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
10
20
30
40
50
60
H.264 Quantization Parameter (QP) ->

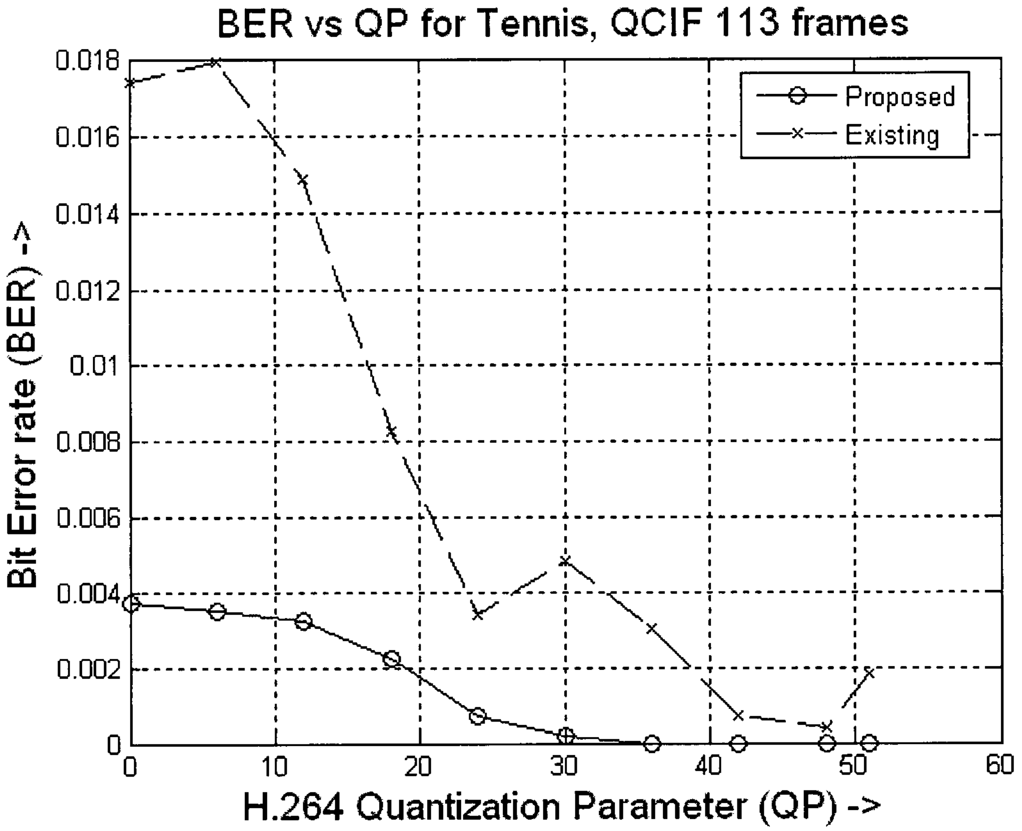
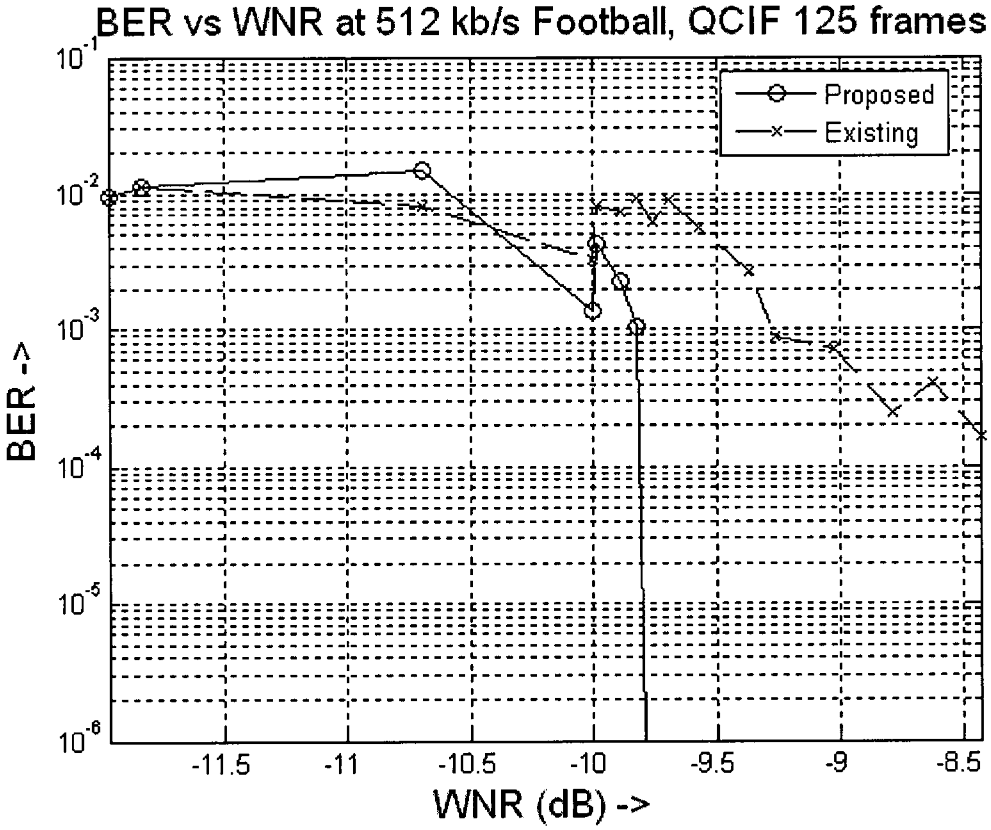
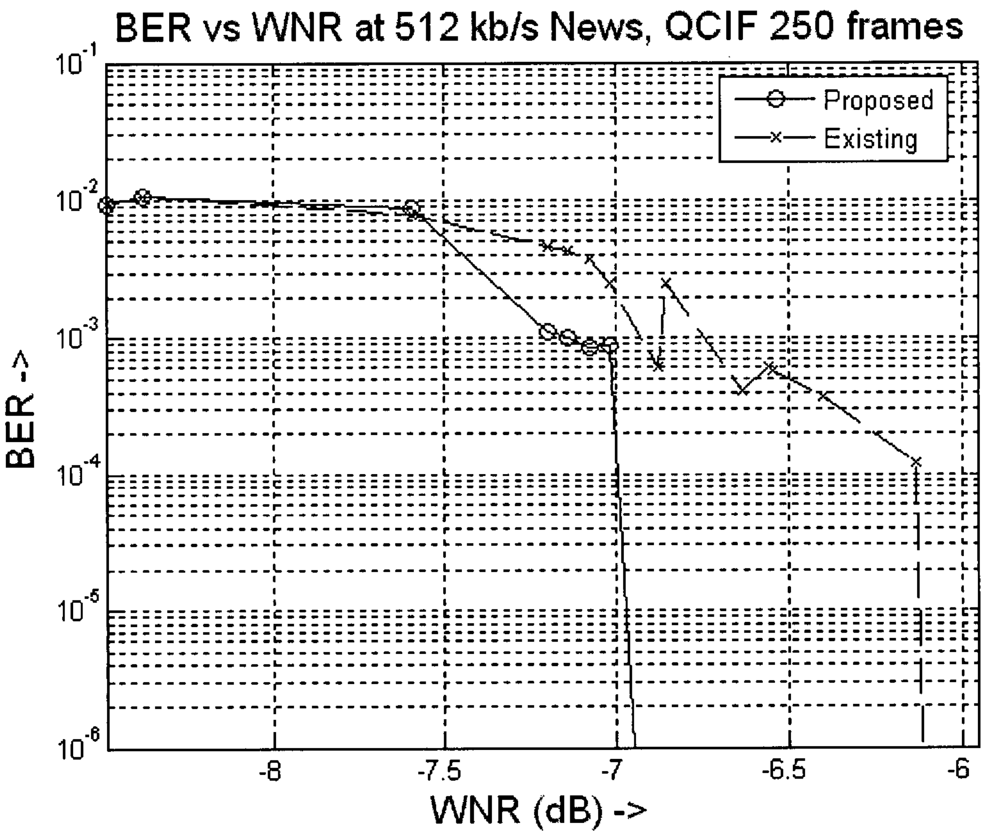


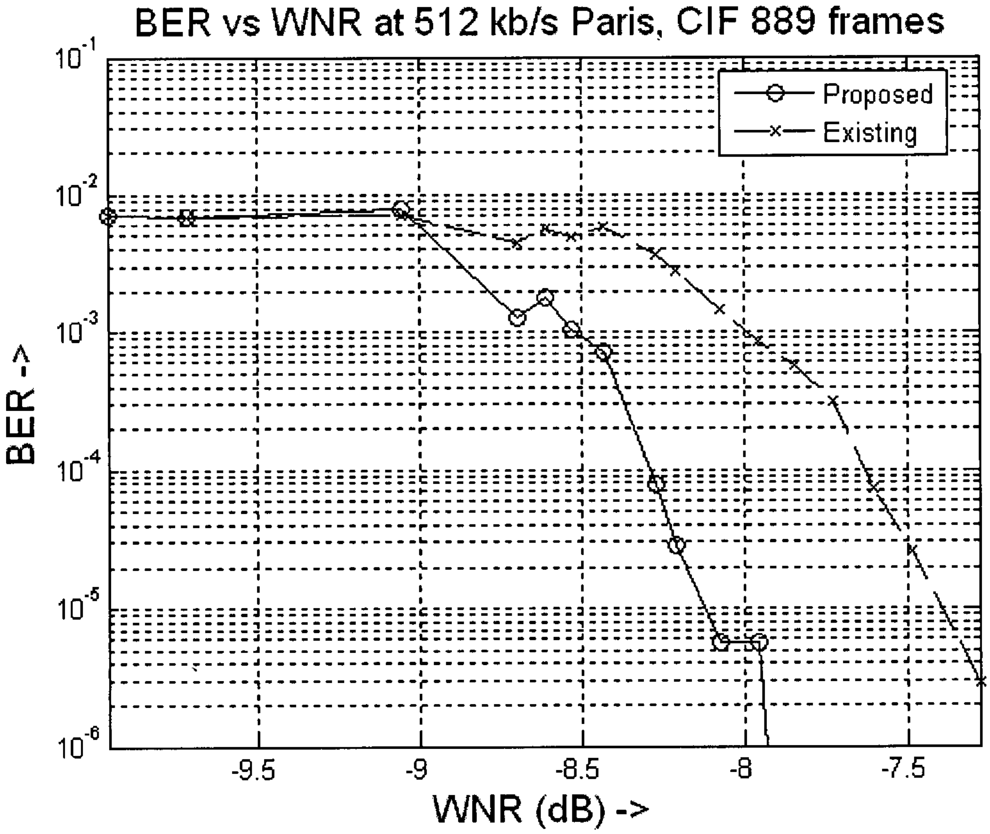
Figure 4.2 Bit error rates after H.264 compression at various bit-rates for four representative watermarked sequences (a) Football (b) News (c) Paris and (d) Tennis

4.2 H.264 compression and decompression different Watermark-to-Noise Ratios (WNRs)

In the second category, the sequences were watermarked at different Watermark-to-Noise Ratios (WNRs), while the video bit-rate was fixed at 512 kb/s. The following plots show BERs for our scheme and ST-SCS. In case of "Football", our scheme requires about 3dB less WNR than ST-SCS, in order to achieve minimum BER. This is because our scheme manages to embed the watermark data much more effectively due to the perceptual mask and the locally optimum watermark scale factor. In case of "News", our scheme requires about 1dB less WNR than ST-SCS. This reduction in the improvement is because "News" only has a moderate level of spatial activity, thus making it harder to embed the watermark without affecting perceptual quality. On an average, our scheme requires about 2dB less WNR in order to achieve minimum BER for the 10 sequences. Also, for the same WNR, the BER achieved by our scheme is about 2 orders of magnitude lower than ST-SCS.




BER vs WNR at 512 kb/s News, QCIF 250 frames
BER ->
WNR (dB) ->
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
-8
-7.5
-7
-6.5
-6
Proposed
Existing


BER vs WNR at 512 kb/s Paris, CIF 889 frames
BER ->
WNR (dB) ->
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
-9.5
-9
-8.5
-8
-7.5
Proposed
Existing

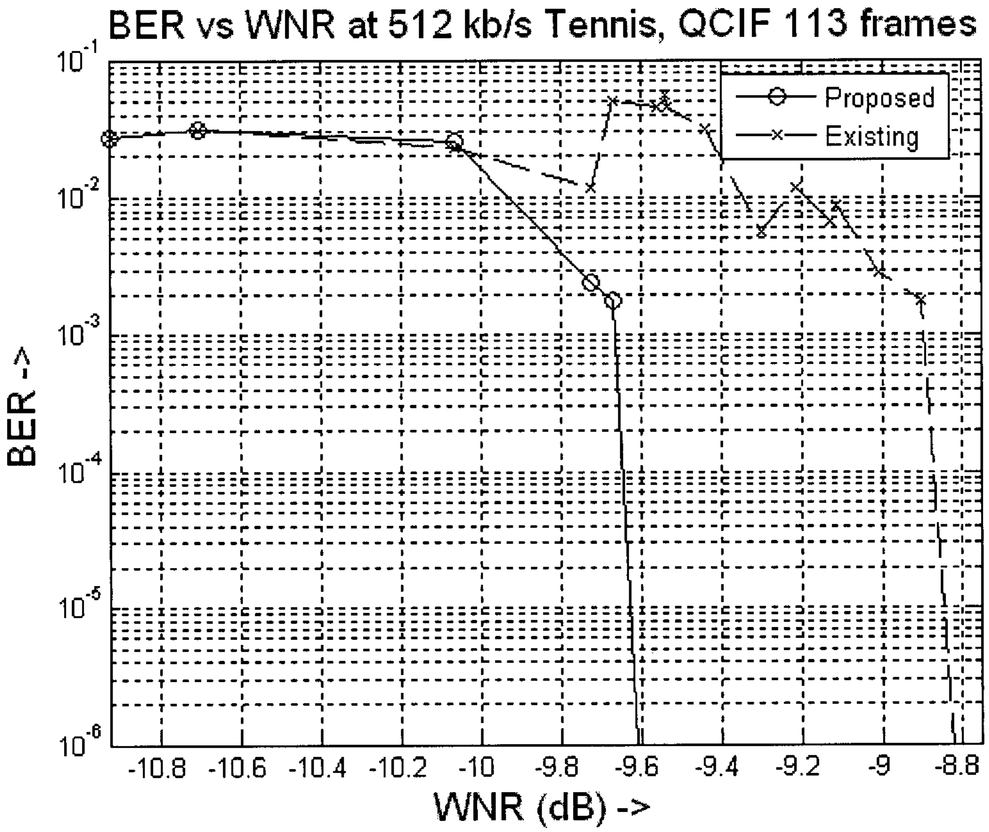


Figure 4.3 Bit error rates after H.264 compression at various WNRs for four representative watermarked sequences (a) Football (b) News (c) Paris and (d) Tennis

4.3 Transcoding attack

In Table 4.2, the results for transcoding attack are shown. For this attack, the video sequences were first watermarked at the given bit-rate, with an Intra period of 5 frames. Next, the sequences were decompressed and recompressed at the same bit-rate, but with a different GOP structure (Intra period = 10). This attack significantly changes the spatial and temporal residual data, when compared to the original compressed stream. Results for 4 representative sequences are tabulated in Table I for 128, 384 and 768 kb/s. In case of "Football", our scheme achieves a BER which is 8 times lower than ST-SCS. On an average, the BER improvements for the 10 sequences shown are about 3 times lower than the corresponding BER achieved by ST-SCS.

Table 4.2 TRANSCODING ATTACK

	Proposed scheme			Existing scheme		
Sequence	**Bit-rate (kb/s)**	**PSNR (dB)**	**BER x 10^{-3}**	**Bit-rate (kb/s)**	**PSNR (dB)**	**BER x 10^{-3}**
Carphone	128	33.9475	3.8752	128	34.1665	10.5139
	384	41.9825	6.6387	384	40.4495	63.9095
	768	46.6450	30.3824	768	46.5110	40.4834
	Average	**40.8583**	**13.6321**		**40.3757**	**38.3023**
Coastguard	128	34.9825	4.1616	128	34.5970	13.9192
	384	37.9905	2.4041	384	35.5875	69.7980
	768	42.5200	5.4141	768	42.1130	52.2828
	Average	**38.4977**	**3.9933**		**37.4325**	**45.3333**
Football	128	26.5770	5.2121	128	26.4825	28.2020
	384	30.3005	10.2626	384	30.8525	46.7475
	768	35.8705	4.2424	768	34.5555	70.7879
	Average	**30.9160**	**6.5724**		**30.6302**	**48.5791**
Foreman	128	32.1010	4.0530	128	33.5645	9.7854
	384	40.4145	4.3056	384	38.5480	65.8586
	768	44.6615	23.0934	768	44.4135	41.8560
	Average	**39.0590**	**10.4840**		**38.8420**	**39.1667**
Flower Garden	128	26.5730	5.7532	128	26.3030	27.5362
	384	29.9035	12.4286	384	31.1355	38.8669
	768	36.0870	2.2398	768	34.8410	70.9267
	Average	**30.8545**	**6.8072**		**30.7598**	**45.7766**
Mother Daughter	128	33.7265	6.2626	128	36.6970	15.1111
	384	47.0670	26.7677	384	46.9085	34.3838
	768	50.6585	35.7778	768	50.5450	45.9798
	Average	**43.8173**	**22.9360**		**44.7168**	**31.8249**
News	128	33.9495	4.2829	128	33.7465	14.5050
	384	45.5590	8.7475	384	44.4605	50.8283
	768	50.7900	33.8586	768	50.6510	45.0707
	Average	**43.4328**	**15.6296**		**42.9527**	**36.8014**
Paris	128	32.9835	2.9939	128	33.0845	5.6811
	384	41.5175	2.5452	384	37.5255	66.4974
	768	48.7240	42.4606	768	48.6150	46.4260
	Average	**41.0750**	**15.9999**		**39.7417**	**39.5348**
Tempete	128	31.7880	3.5073	128	31.2710	15.8763
	384	35.6055	4.7699	384	32.4690	74.7054
	768	40.7680	3.6242	768	40.1895	60.6528
	Average	**36.0538**	**3.9671**		**34.6432**	**50.4115**
Tennis	128	35.0400	10.0563	128	34.5145	35.2977
	384	41.4275	14.4811	384	40.1875	69.9920
	768	45.9150	35.1468	768	45.6350	52.9968
	Average	**40.7942**	**19.8948**		**40.1123**	**52.7621**
Overall		**38.5359**	**11.9916**		**38.0207**	**42.8493**

4.4 Gaussian low-pass filtering attack

Table 4.3 summarizes the 3x3 Gaussian low-pass filtering attacks on both watermarking schemes. It can be observed that our scheme shows superior performance on the "Football" sequence, in which case it achieves a BER which is 1/10th of the bit-error rate of ST-SCS. For sequences having less spatial activity, the BER of our scheme is less than half that of the corresponding ST-SCS value.

Table 4.3 3X3 GAUSSIAN LOW-PASS FILTERING ATTACK

	Proposed scheme			Existing scheme		
Sequence	**Bit-rate (kb/s)**	**PSNR (dB)**	**BER x 10^{-3}**	**Bit-rate (kb/s)**	**PSNR (dB)**	**BER x 10^{-3}**
Carphone	128	18.840	2.5094	128	18.347	9.2910
	384	39.352	7.0834	384	37.521	60.8443
	768	44.755	64.3383	768	44.562	76.3769
	Average	**34.3157**	**24.6437**		**33.4767**	**48.8374**
Coastguard	128	18.567	4.1616	128	18.304	13.9192
	384	34.975	2.4041	384	32.075	69.7980
	768	40.321	5.4141	768	39.792	52.2828
	Average	**31.2877**	**3.9933**		**30.0570**	**45.3333**
Football	128	20.051	1.4141	128	19.600	23.0707
	384	25.001	4.9293	384	22.356	34.4242
	768	33.224	6.0202	768	31.476	64.3232
	Average	**26.0920**	**4.1212**		**24.4773**	**40.6060**
Foreman	128	20.360	3.0555	128	19.769	8.3838
	384	37.394	2.9419	384	35.302	62.0580
	768	42.405	54.0025	768	42.104	78.3964
	Average	**33.3863**	**20.0000**		**32.3917**	**49.6128**
Flower Garden	128	20.214	1.5371	128	19.677	23.6715
	384	24.248	2.4154	384	21.668	24.9012
	768	33.350	22.3715	768	31.667	86.2275
	Average	**25.9373**	**8.7747**		**24.3373**	**44.9334**
Mother Daughter	128	21.583	4.8283	128	20.536	12.6060
	384	44.667	33.1515	384	44.522	41.2323
	768	48.885	72.2222	768	48.679	81.0909
	Average	**38.3783**	**36.7340**		**37.9123**	**44.9764**
News	128	18.817	3.0707	128	18.543	11.6161
	384	42.483	14.1208	384	41.470	54.3232
	768	48.777	62.6464	768	48.426	72.9697
	Average	**36.6923**	**26.6126**		**36.1463**	**46.3030**
Paris	128	16.077	2.0565	128	15.900	4.4142
	384	37.999	4.0563	384	34.079	63.0376
	768	46.156	73.2181	768	45.979	76.8710
	Average	**33.4107**	**26.4437**		**31.9860**	**48.1076**
Tempete	128	18.719	0.9586	128	18.419	13.0939
	384	32.375	5.0739	384	28.680	64.6511
	768	38.372	25.2993	768	37.493	83.6606
	Average	**29.8220**	**10.4439**		**28.1973**	**53.8019**
Tennis	128	22.605	6.3354	128	21.981	27.9565
	384	38.326	11.5647	384	36.837	64.9135
	768	43.769	77.3330	768	43.460	94.6299
	Average	**34.9**	**31.7444**		**34.0927**	**62.5000**
Overall		**32.4223**	**19.3511**		**31.3074**	**48.5012**

4.5 Downscaling attack with bilinear sampling

In Table 4.4, the 75% downscaling attacks are summarized for 4 streams. Bilinear sampling was used for the scaling operation. On an average, our scheme yields a BER that is more than 2 times lower than that of ST-SCS. In fact for “Football”, the improvement is about 20 times compared to ST-SCS.

Table 4.4 75% DOWNSCALING WITH BILINEAR SAMPLING

	Proposed scheme			Existing scheme		
Sequence	**Bit-rate (kb/s)**	**PSNR (dB)**	**BER x 10^{-3}**	**Bit-rate (kb/s)**	**PSNR (dB)**	**BER x 10^{-3}**
Carphone	128	29.0595	1.4611	128	28.9660	8.6557
	384	42.3365	2.5729	384	40.8210	57.0008
	768	47.2545	36.1000	768	47.1285	46.2806
	Average	**39.5502**	**13.3780**		**38.9718**	**37.3124**
Coastguard	128	29.7310	1.0303	128	29.4875	11.0303
	384	38.5935	2.0404	384	36.1695	62.4242
	768	43.4575	5.6768	768	43.0675	51.2727
	Average	**37.2607**	**2.9158**		**36.2415**	**41.5758**
Football	128	24.6380	0.6061	128	24.1825	22.4242
	384	30.9530	3.8384	384	30.1050	33.5757
	768	37.3665	1.6970	768	36.0410	60.6464
	Average	**30.9858**	**2.0471**		**30.1095**	**38.8821**
Foreman	128	29.1990	2.2727	128	29.0000	7.8535
	384	40.8225	1.7424	384	39.0255	61.5783
	768	45.4735	27.5759	768	45.2530	46.7679
	Average	**38.4983**	**10.5304**		**37.7595**	**38.7332**
Flower Garden	128	24.5480	0.6149	128	23.8850	22.5736
	384	30.3820	2.8546	384	30.1025	25.1208
	768	37.4880	1.5810	768	36.1770	63.6364
	Average	**30.8060**	**1.6835**		**30.0548**	**37.1102**
Mother Daughter	128	30.7430	4.9899	128	31.2270	12.9697
	384	47.2870	24.4646	384	47.1540	31.7172
	768	51.0225	44.9293	768	50.9290	53.8181
	Average	**43.0175**	**24.7946**		**43.1033**	**32.8350**
News	128	29.1530	1.5960	128	28.8200	11.4747
	384	45.5480	7.4546	384	44.5095	49.4343
	768	51.0715	41.0907	768	50.9015	52.8285
	Average	**41.9242**	**16.7137**		**41.4103**	**37.9125**
Paris	128	27.2625	1.1249	128	27.3795	3.8348
	384	41.5440	0.4999	384	37.7385	63.2307
	768	48.7925	51.6471	768	48.6920	55.3736
	Average	**39.1997**	**17.7573**		**37.9367**	**40.8130**
Tempete	128	27.7865	1.1223	128	27.1555	13.2108
	384	36.4075	2.8058	384	33.3950	64.4875
	768	42.0650	3.4372	768	41.3930	57.6599
	Average	**35.4197**	**2.4551**		**33.9812**	**45.1194**
Tennis	128	35.2550	8.1958	128	34.6905	29.4650
	384	44.2060	13.4753	384	42.9885	66.3717
	768	49.2400	73.5111	768	48.9820	92.8710
	Average	**42.9003**	**31.7274**		**42.2203**	**62.9026**
Overall		**37.9562**	**12.4003**		**37.1789**	**41.3196**

4.6 Rotation attack with bilinear sampling

Table 4.5 shows BER results for the 5° counter-clockwise rotation attack with bilinear sampling. The results obtained are similar to that for downscaling attack. On an average, our scheme achieves a BER that is less than half of that of ST-SCS.

Table 4.5 5° ROTATION WITH BILINEAR SAMPLING

	Proposed scheme			**Existing scheme**		
Sequence	**Bit-rate (kb/s)**	**PSNR (dB)**	**BER x 10^{-3}**	**Bit-rate (kb/s)**	**PSNR (dB)**	**BER x 10^{-3}**
Carphone	128	32.4040	2.7317	128	32.5350	9.1640
	384	43.6605	5.4793	384	42.2155	59.0020
	768	48.4570	55.7305	768	48.3305	66.9271
	Average	**41.5072**	**21.3138**		**41.0270**	**45.0310**
Coastguard	128	33.8460	2.6263	128	33.6275	11.8384
	384	40.7530	2.3031	384	38.4150	63.9798
	768	45.5515	23.5556	768	45.1670	74.8889
	Average	**40.0502**	**9.4950**		**39.0698**	**50.2357**
Football	128	28.5860	1.8990	128	27.8005	23.5959
	384	33.1200	5.5354	384	32.5375	34.3030
	768	39.7035	5.9799	768	38.5150	62.7879
	Average	**33.8032**	**4.4714**		**32.9510**	**40.2290**
Foreman	128	32.2440	3.3207	128	32.4435	8.6869
	384	42.0985	2.5000	384	40.4045	61.7424
	768	46.6795	43.1564	768	46.4595	65.2779
	Average	**40.3407**	**16.3257**		**39.7692**	**45.2357**
Flower Garden	128	29.7435	1.7128	128	28.8720	23.8472
	384	32.9925	2.3276	384	32.9490	25.6478
	768	39.7400	3.5134	768	38.5400	65.6127
	Average	**34.1587**	**2.5179**		**33.4537**	**38.3692**
Mother Daughter	128	33.7380	5.2929	128	35.3025	12.8687
	384	48.4115	30.9292	384	48.2580	38.3639
	768	52.5335	70.0018	768	52.4300	80.2206
	Average	**44.8943**	**35.4079**		**45.3302**	**43.8177**
News	128	32.0565	3.1515	128	31.8580	12.5050
	384	46.9700	13.6568	384	45.8785	53.4344
	768	52.8655	65.5152	768	52.7015	76.6265
	Average	**43.9640**	**27.4411**		**43.4793**	**47.5220**
Paris	128	30.5190	2.1247	128	30.7320	4.5051
	384	43.3755	4.0279	384	39.4385	63.4125
	768	50.5625	74.1216	768	50.4590	78.6661
	Average	**41.4857**	**26.7581**		**40.2098**	**48.8612**
Tempete	128	31.6595	1.3328	128	31.1800	13.4212
	384	37.8535	5.0504	384	35.2135	64.4875
	768	43.7210	16.5544	768	43.0550	72.2968
	Average	**37.7447**	**7.6459**		**36.4828**	**50.0685**
Tennis	128	31.7040	5.7323	128	31.0265	26.5487
	384	42.0270	8.4973	384	40.8080	62.9022
	768	46.8560	38.6161	768	46.6315	58.6785
	Average	**40.1957**	**17.6153**		**39.4887**	**49.3765**
Overall		**39.8144**	**16.8992**		**39.1261**	**45.8747**

4.7 Averaging Collusion attack

In Table 4.6, the results for Collusion attack on both watermarking schemes are tabulated. For this attacks, 5 copies of a given video sequence are compressed at 512kb/s and watermarked using different keys (hence resulting in 5 different watermark sequences). Next, the 5 watermarked video sequences are averaged in order to obtain a 6th video sequence. Then, using the 5 original watermark keys, the watermark is extracted from the colluded video sequence. The average BER obtained by this process is tabulated for both schemes. On an average, our scheme achieves a BER which is less than 1/4th that of ST-SCS, after collusion.

Table 4.6 COLLUSION ATTACK

	Proposed scheme			**Existing scheme**		
Sequence	**Bit-rate (kb/s)**	**PSNR (dB)**	**BER x 10^{-3}**	**Bit-rate (kb/s)**	**PSNR (dB)**	**BER x 10^{-3}**
Carphone	512	39.3520	17.057	512	37.5210	63.242
Coastguard	512	34.9750	7.6768	512	32.0750	77.9596
Football	512	25.0010	5.1313	512	22.3560	67.475
Foreman	512	37.3940	5.9343	512	35.3020	68.1818
Flower Garden	512	24.2480	4.8309	512	21.6680	66.4032
Mother Daughter	512	44.6670	29.1515	512	44.5220	34.7475
News	512	42.4830	31.010	512	41.4700	47.131
Paris	512	37.9990	8.3967	512	34.0790	69.554
Tempete	512	32.3750	14.3799	512	28.6800	78.5400
Tennis	512	38.3260	21.269	512	36.8370	81.255
Overall		**35.6820**	**14.4837**		**33.4510**	**65.4489**

4.8 Watermarked video visual quality comparison

It is well known that PSNR alone is not a good measure of perceptual quality ([25], [26], [27]). Hence, in order to measure the visual quality of the watermarked video, we use the Structural Similarity Index (SSIM) [28]. The SSIM score is given on a scale of 0-1, by comparing the watermarked video with the unwatermarked video. A higher value indicates that the resulting video is perceptually closer to the original sequence. The visual quality results for both watermarking schemes is tabulated in Table 4.7. It shows that our scheme always maintains a higher perceptual quality score over ST-SCS.

Table 4.7 VISUAL QUALITY COMPARISON

	Proposed scheme		**Existing scheme**	
Sequence	**Bit-rate (kb/s)**	SSIM (0-1)	**Bit-rate (kb/s)**	SSIM (0-1)
Carphone	128	0.59388	128	0.57988
	384	0.97782	384	0.96439
	768	0.99113	768	0.9909
	Average	**0.85428**		**0.84506**
Coastguard	128	0.49323	128	0.4856
	384	0.94258	384	0.90732
	768	0.97959	768	0.97863
	Average	**0.80513**		**0.79052**
Football	128	0.54299	128	0.52896
	384	0.74624	384	0.64665
	768	0.93665	768	0.91701
	Average	**0.74196**		**0.69754**
Foreman	128	0.61817	128	0.60162
	384	0.96643	384	0.95037
	768	0.98636	768	0.98595
	Average	**0.85699**		**0.84598**
Flower Garden	128	0.58396	128	0.5692
	384	0.80035	384	0.69381
	768	0.97101	768	0.9599
	Average	**0.7851**		**0.74097**
Mother Daughter	128	0.63674	128	0.63216
	384	0.98952	384	0.98945
	768	0.99493	768	0.99478
	Average	**0.87373**		**0.87213**
News	128	0.56201	128	0.55225
	384	0.98727	384	0.98498
	768	0.99571	768	0.99551
	Average	**0.84833**		**0.84425**
Paris	128	0.46781	128	0.46313
	384	0.9816	384	0.95452
	768	0.99605	768	0.99597
	Average	**0.81515**		**0.80454**
Tempete	128	0.53629	128	0.52589
	384	0.95809	384	0.89729
	768	0.98706	768	0.98483
	Average	**0.82714**		**0.80267**
Tennis	128	0.63542	128	0.6154
	384	0.94445	384	0.93007
	768	0.9806	768	0.97982
	Average	**0.85349**		**0.84176**
Overall		**0.82613**		**0.80854**

5 Conclusions and Future Work

5.1 Conclusions and contributions

In this thesis, we have presented an improved scalar quantization-based digital video watermarking scheme, which is designed to work for the H.264/AVC video codec. The main contributions of this thesis can be summarized as follows:

- The proposed watermarking scheme consists of a locally adaptive, Rate-Distortion optimized watermark which is inserted in the transform coefficients of macroblock residuals. This ensures that watermark signal is embedded in the most robust manner, with least visual distortion. Our scheme adapts to the characteristics of the video signal at the Macroblock level and computes the watermark scale factor based on local statistics.
- We use a unique perceptual mask in order to limit the spatial and temporal distortion caused due to watermark embedding. Therefore, our scheme achieves higher watermarked picture quality compared to existing schemes.
- Our scheme is designed with a built-in bit-rate control mechanism ensures optimum watermark bit allocation. Therefore, the watermark bits are distributed in proportion to the visual importance of different regions of the video frame.
- We have adapted our scheme to H.264/AVC, which is the latest video coding standard. Our scheme overcomes the challenges for watermarking of

H.264/AVC video, namely high compression efficiency, small residual data, integer transform, Rate-Distortion coding decisions and video bit-rate control. Due to these features, our proposed scheme performs significantly better in terms of bit-error rates and perceptual video quality than traditional Spread Transform Scalar Costa Scheme (ST-SCS). Experiments were conducted thoroughly on 10 standard video test sequences. The results obtained are summarized below:

- In the category of H.264 compression and decompression attack at different video bit-rates, our scheme yields bit-error rate improvements of more than two orders of magnitude compared to ST-SCS, at the same picture quality.
- In case of H.264 compression and decompression at a fixed video bit-rate, with different Watermark-to-Noise Ratios (WNRs), our scheme achieves the same Bit Error Rate as ST-SCS, using 2dB less Watermark-to-Noise (WNR) on an average. For the same level of WNR, the BER improvement is more than two orders of magnitude.
- After applying the transcoding attack, by recompressing the watermarked video at the same bit-rate but a different Group-of-Pictures (GOP) structure, our scheme achieves an average BER which is 3 times less than that of ST-SCS.
- After applying a 3x3 Gaussian low-pass filtering attack, our scheme yields a BER that is less than half of ST-SCS, on an average.
- In case of geometric attacks such as 75% downscaling and 5° rotation with bilinear sampling, the BER obtained is less than half of that of ST-SCS.

- In the category of collusion attacks using the average of 5 watermarked copies of a given video sequence, our schemes achieves a BER that is less than one fourth that of traditional ST-SCS.
- The visual quality of our watermarked video was measured using the Structural Similarity Index (SSIM) scale at different video bit-rates and was found to be *always* perceptually better than ST-SCS.

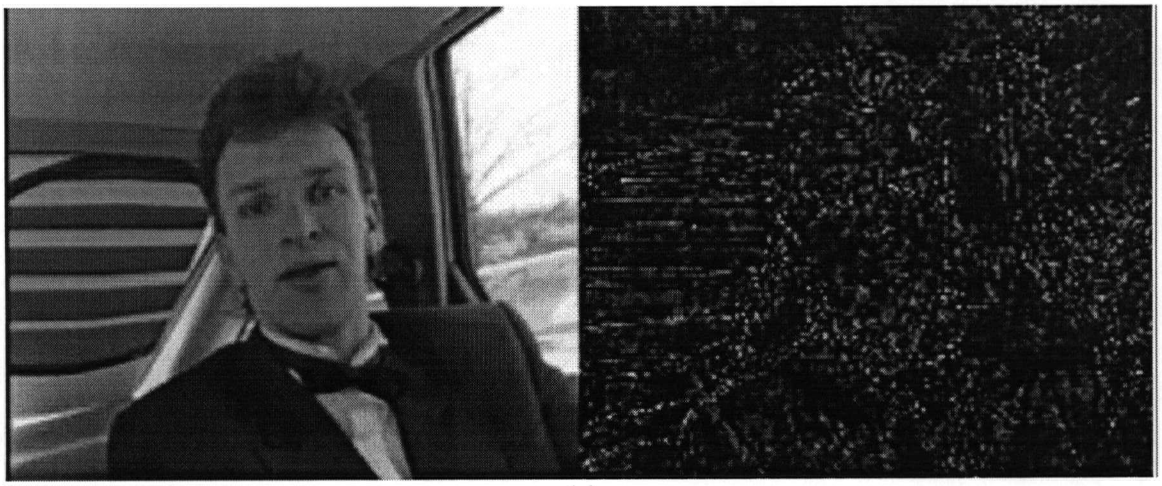
5.2 Future Work

The proposed scheme has been found to work well on several typical video test sequences having varying levels of camera and content motion, as well as spatial detail. The frame sizes for these sequences were QCIF, with a resolution of 176x144 pixels and CIF, with a resolution of 352x288 pixels. These resolutions are commonly used in video streaming and video conferencing applications. In future, more investigation is needed for Standard Definition (720x576 Interlaced) and High Definition (1280x720 Progressive) video content. Such high resolutions are generally used for entertainment quality applications and are encoded using very high bit-rates. The emphasis in this case is picture quality rather than compression efficiency. Therefore, inserting a watermark in a robust manner, without affecting perceptual quality is a challenging task. One possible solution is to embed the watermark into only a few transform coefficients, taking into account the sensitivity of the Human Visual System (HVS).

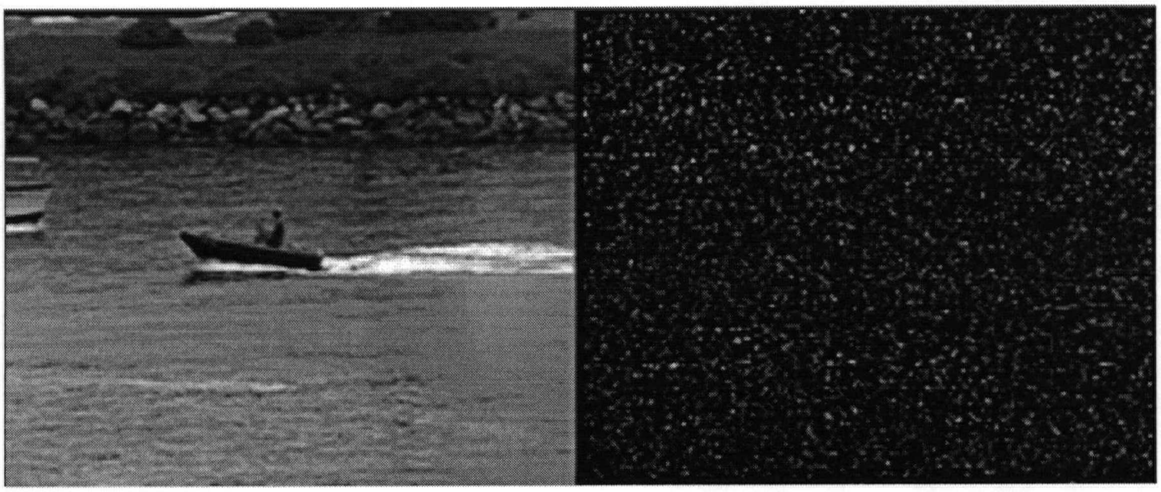
Another aspect which requires further research is to study the category of collusion attacks. We have shown that our scheme performs significantly better than ST-SCS after a simple collusion attack, comprising of an averaging attack with five watermarked sequences. However, previous research on collusion attacks has shown that the robustness to collusion decreases with the increase in the number of watermarked copies available to the malicious user. Besides, advanced users may apply more advanced attacks in order to remove the watermark. Therefore, it is desirable to include collusion resistant design features in the watermarking scheme itself ([11], [12], [30]).

Appendix

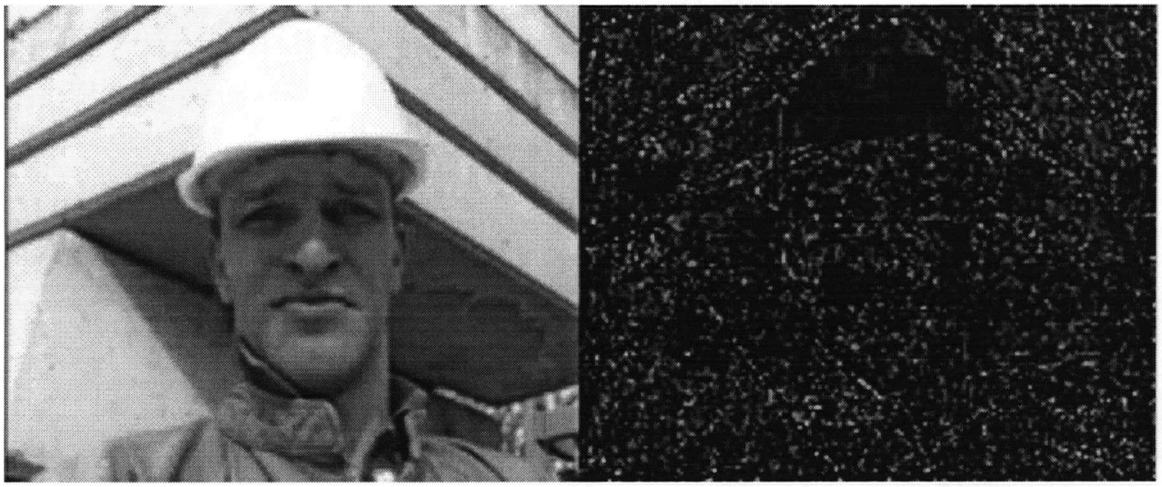
A.1 Watermarked frames and watermark sequences using proposed scheme



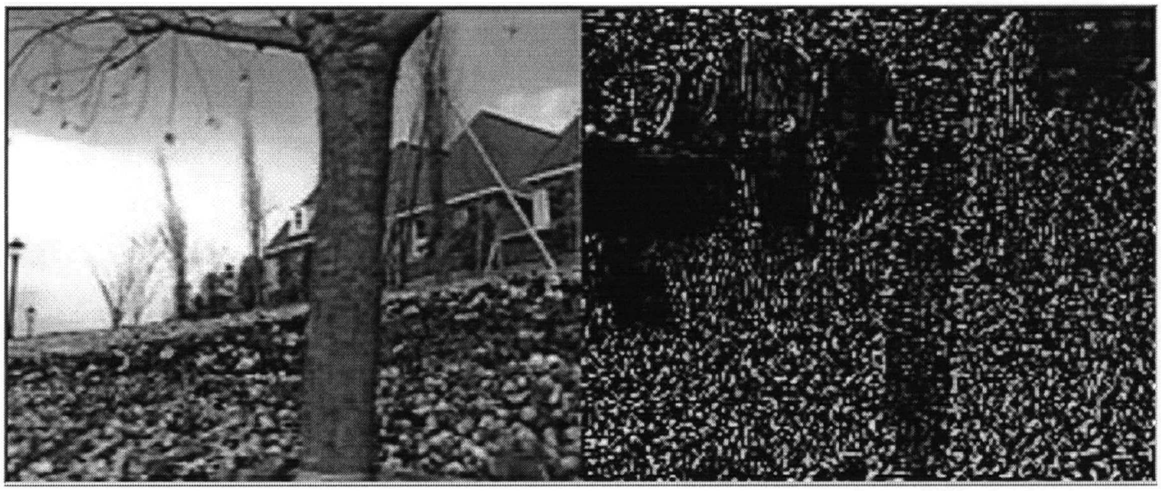
(a)



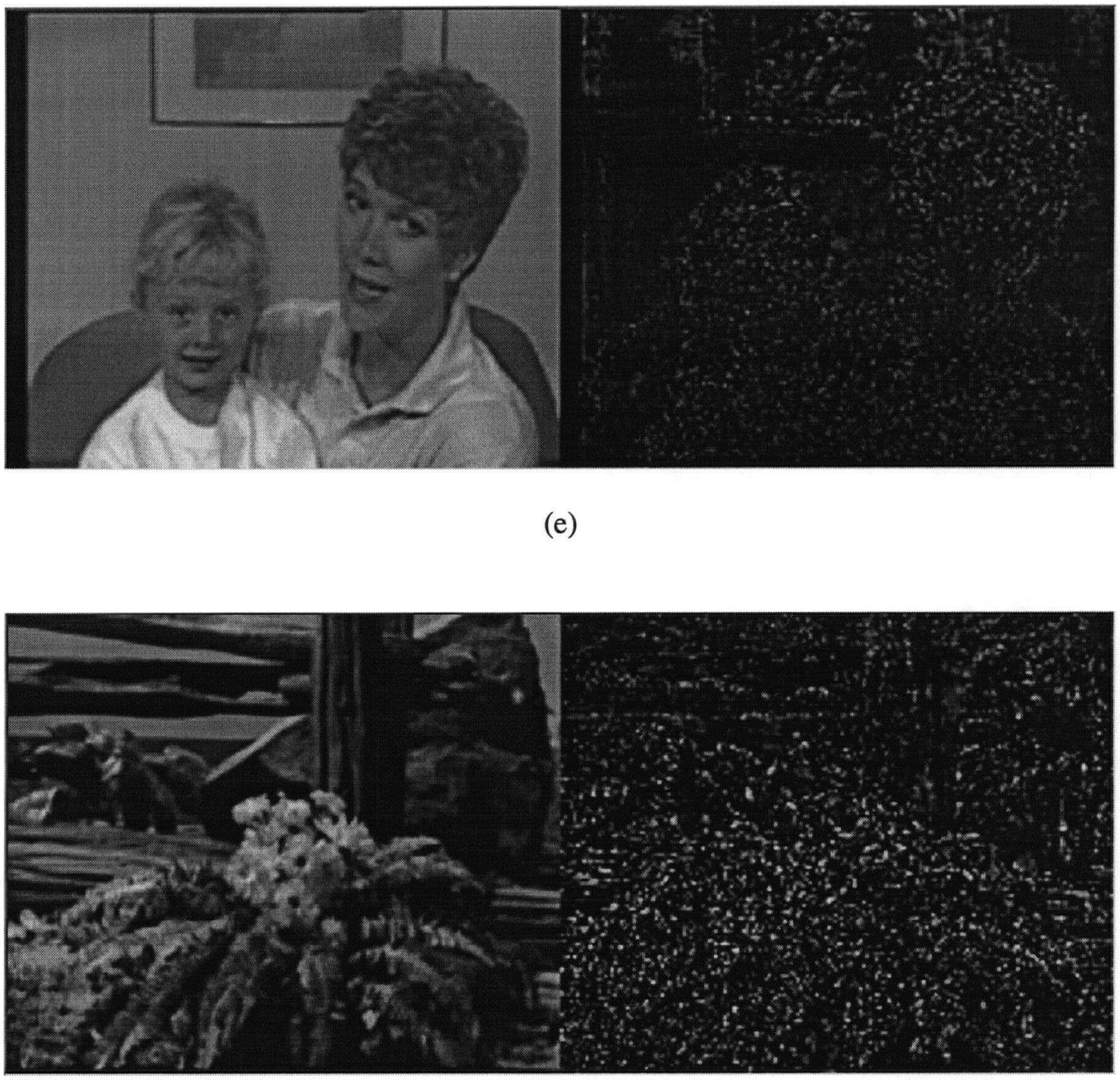
(b)



(c)



(d)

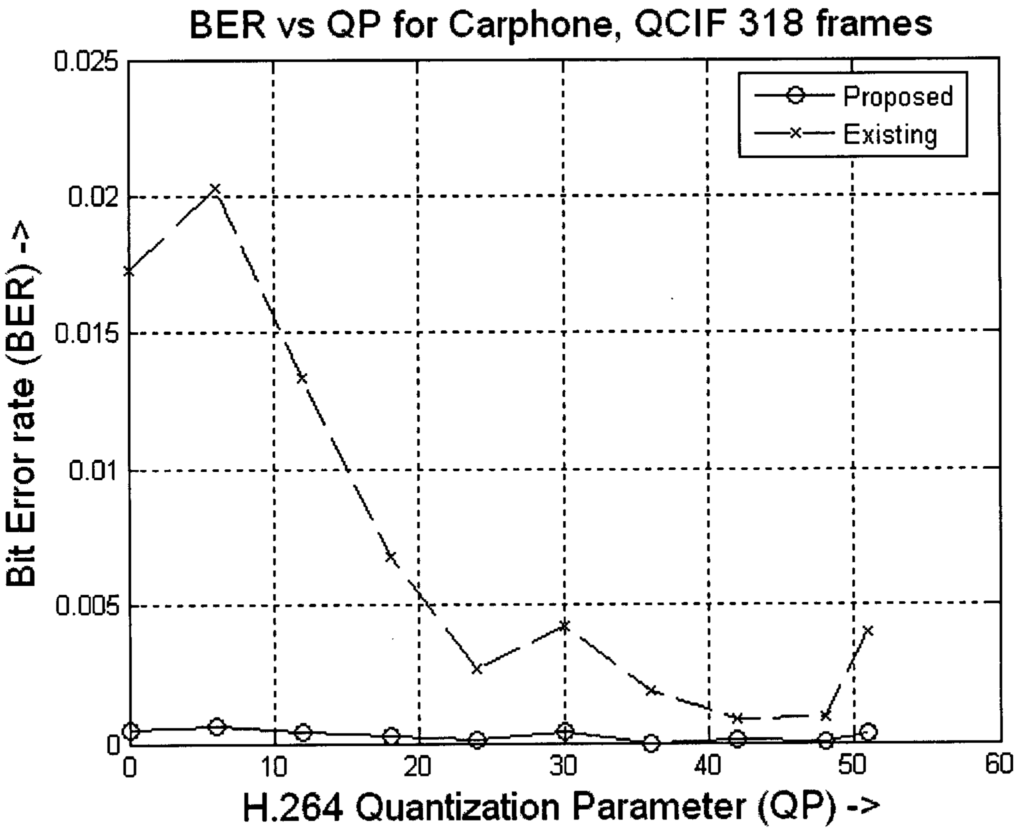


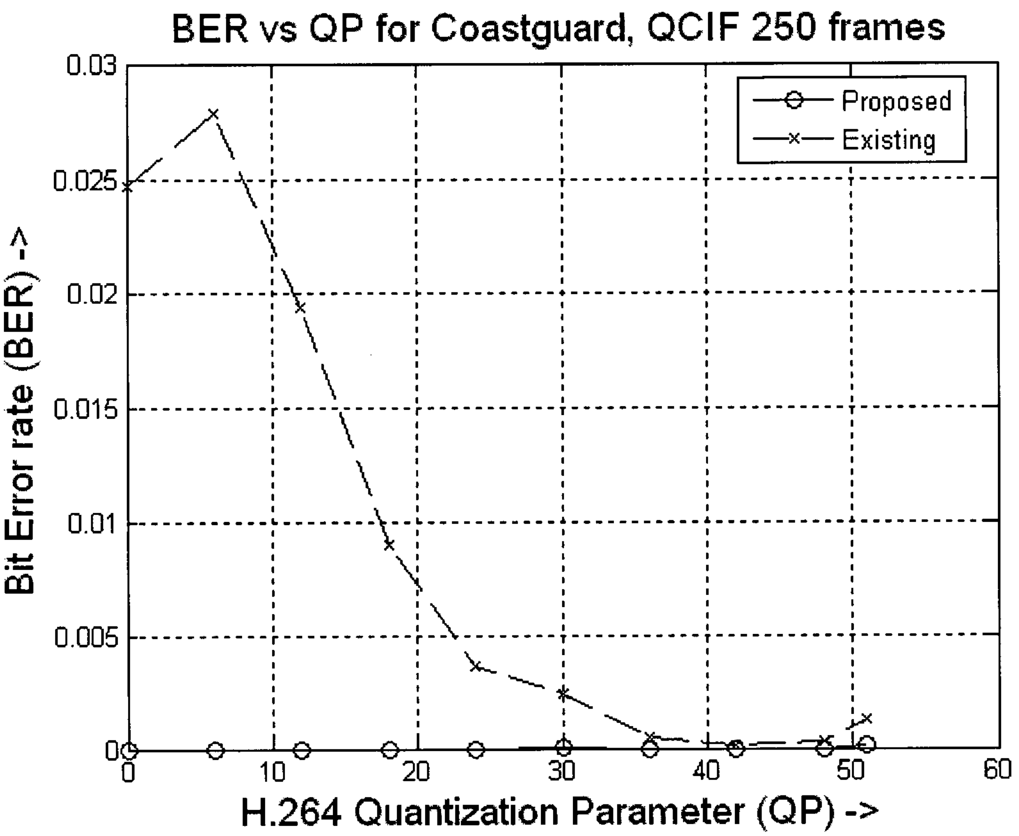
(e)

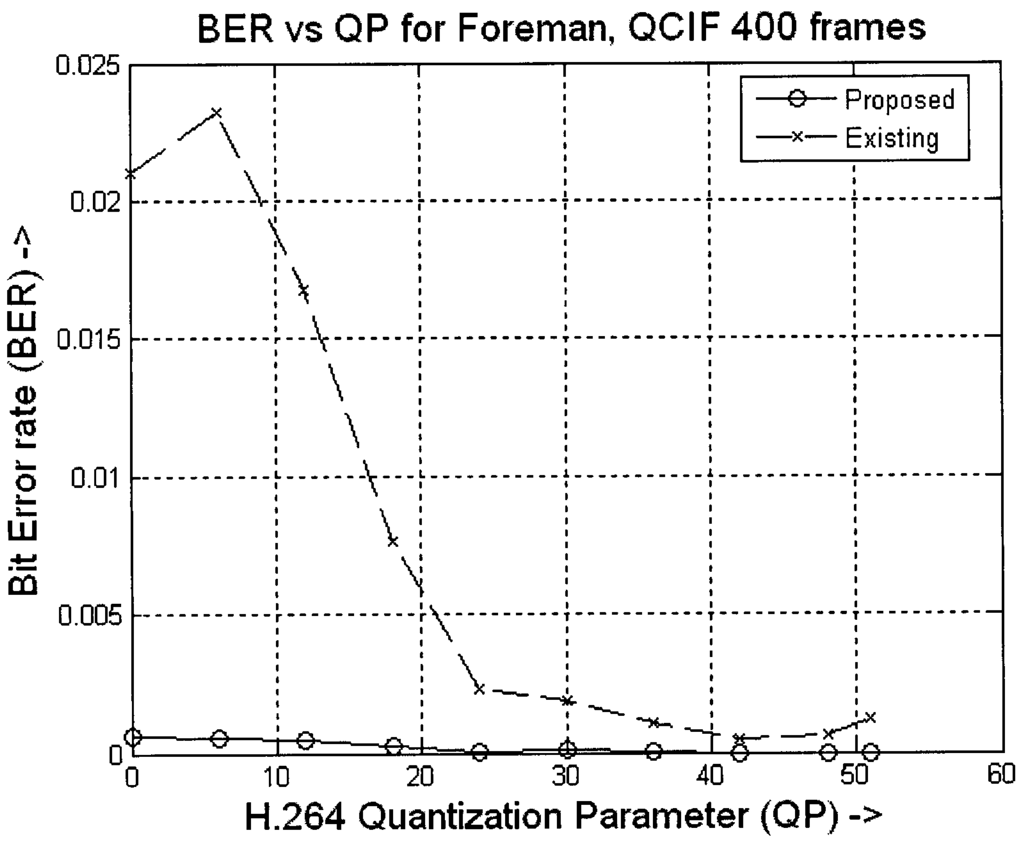
(f)

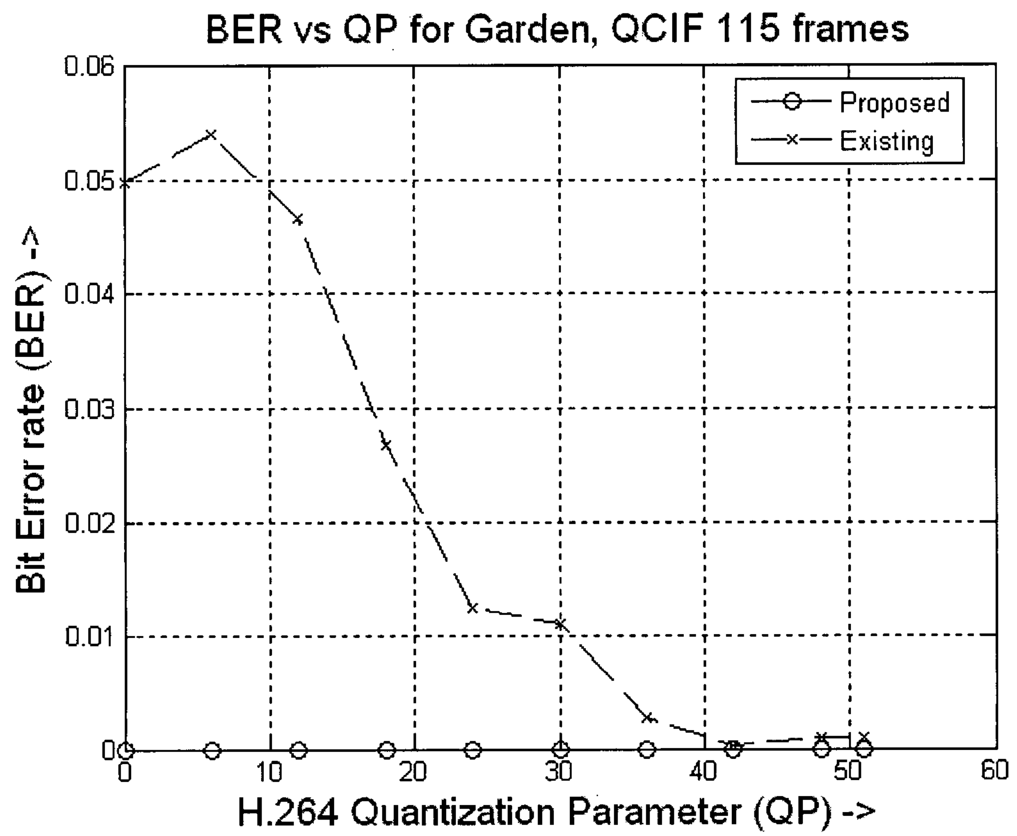
Figure A.1 Watermarked frame (L) and the corresponding watermark sequences for (a) Carphone (b) Coastguard (c) Foreman (d) Flower Garden (e)Mother Daughter and (f) Tempete sequences

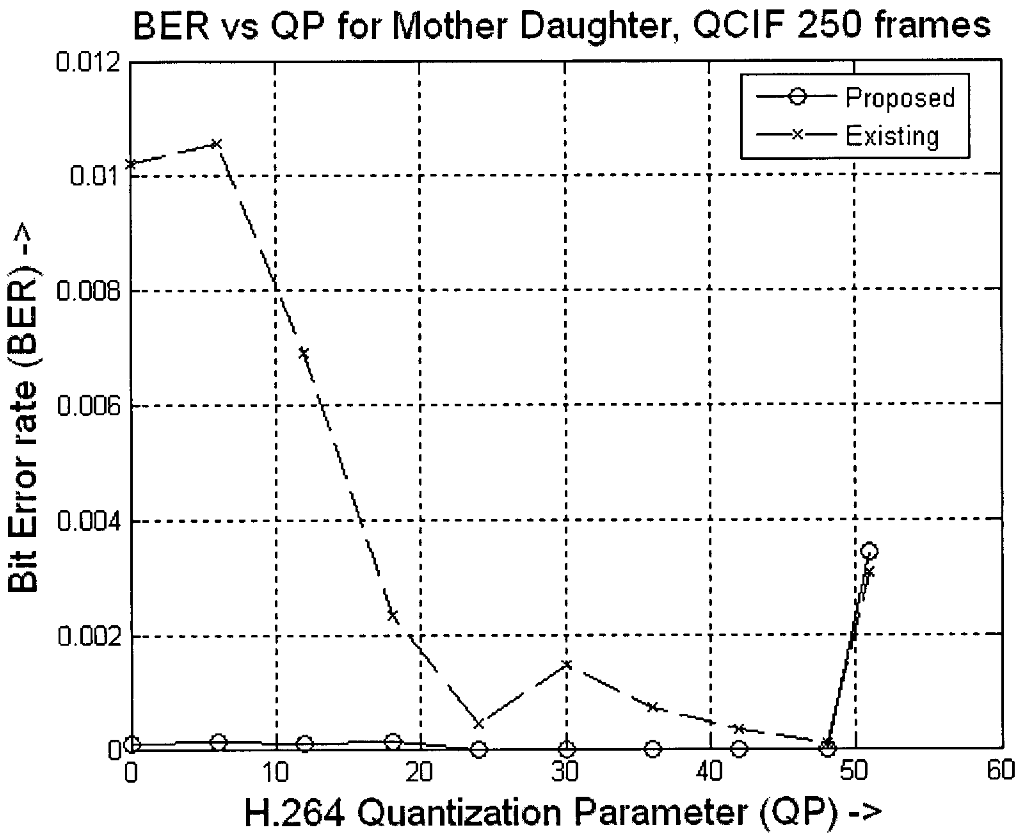
A.2 BER plots for H.264 compression and decompression attack at different bit-rates




BER vs QP for Coastguard, QCIF 250 frames
Bit Error rate (BER) ->
H.264 Quantization Parameter (QP) ->
0.03
0.025
0.02
0.015
0.01
0.005
0
0
10
20
30
40
50
60
Proposed
Existing


BER vs QP for Foreman, QCIF 400 frames
Bit Error rate (BER) ->
H.264 Quantization Parameter (QP) ->
0.025
0.02
0.015
0.01
0.005
0
0
10
20
30
40
50
60
Proposed
Existing


BER vs QP for Garden, QCIF 115 frames
Bit Error rate (BER) ->
0.06
0.05
0.04
0.03
0.02
0.01
0
0
10
20
30
40
50
60
H.264 Quantization Parameter (QP) ->
Proposed
Existing


BER vs QP for Mother Daughter, QCIF 250 frames
Bit Error rate (BER) ->
0.012
0.01
0.008
0.006
0.004
0.002
0
0
10
20
30
40
50
60
H.264 Quantization Parameter (QP) ->
Proposed
Existing

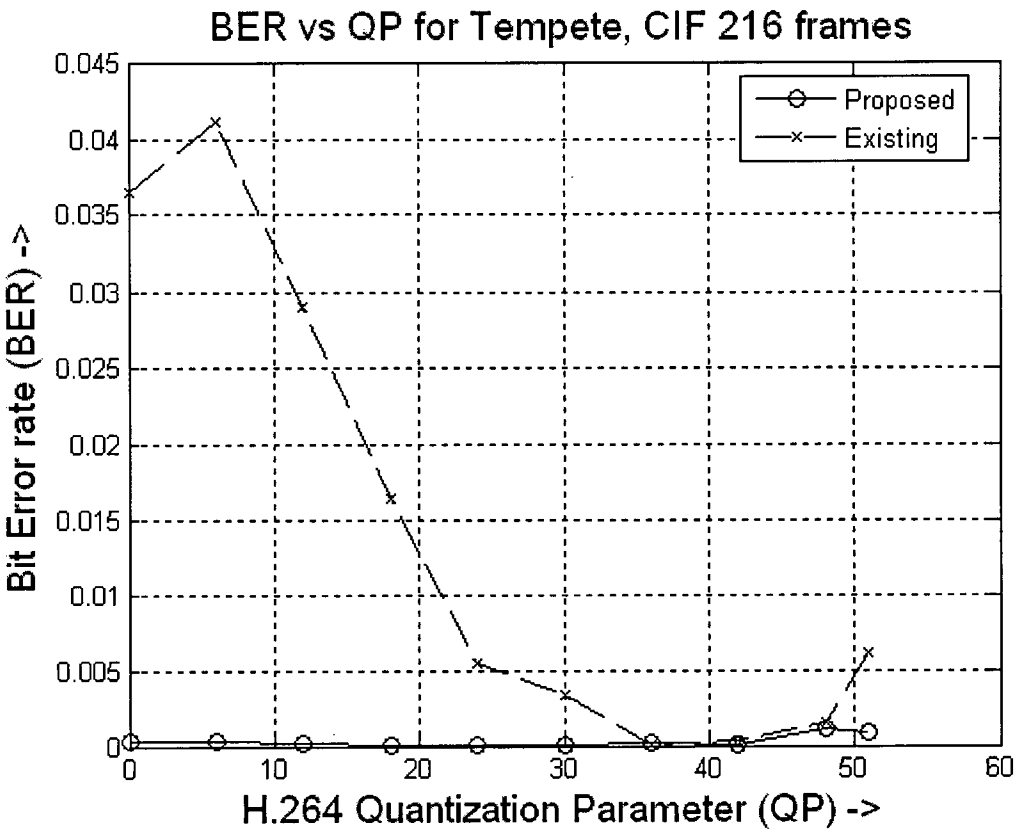
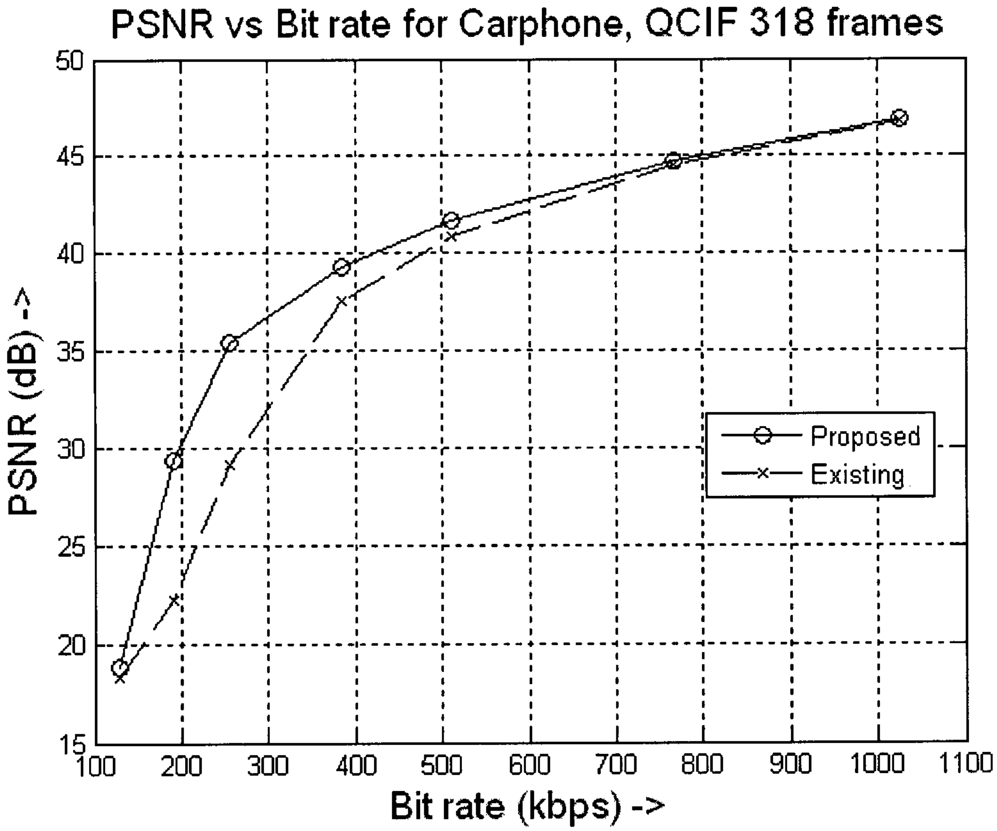
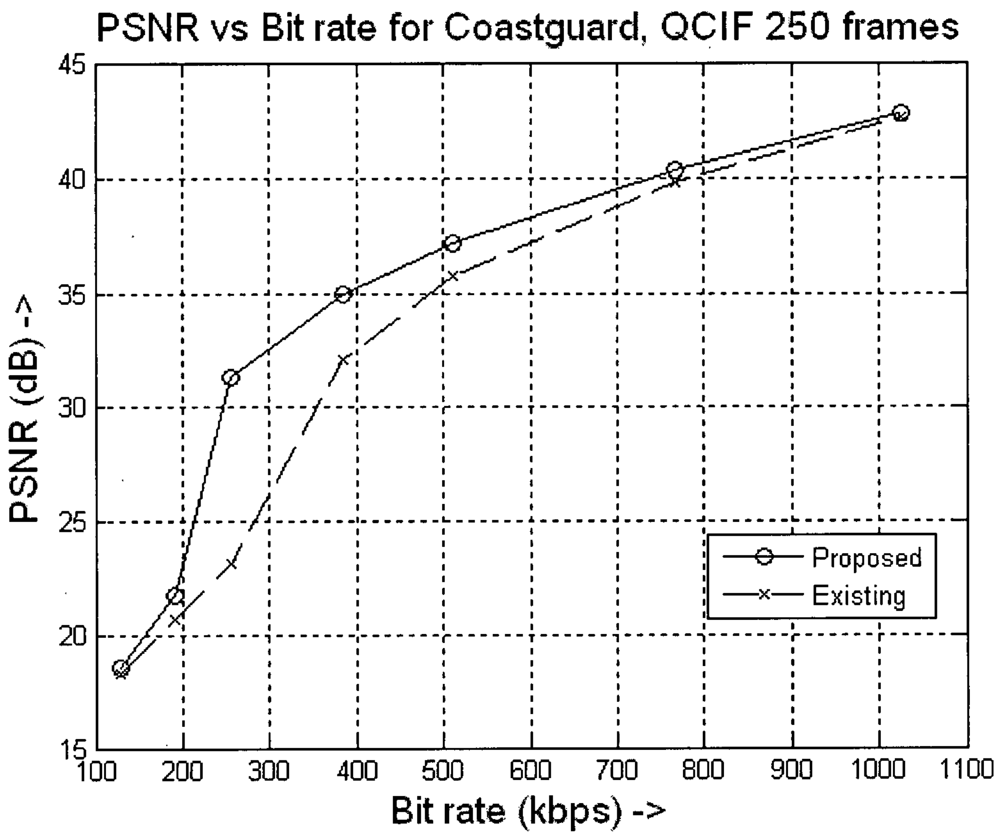
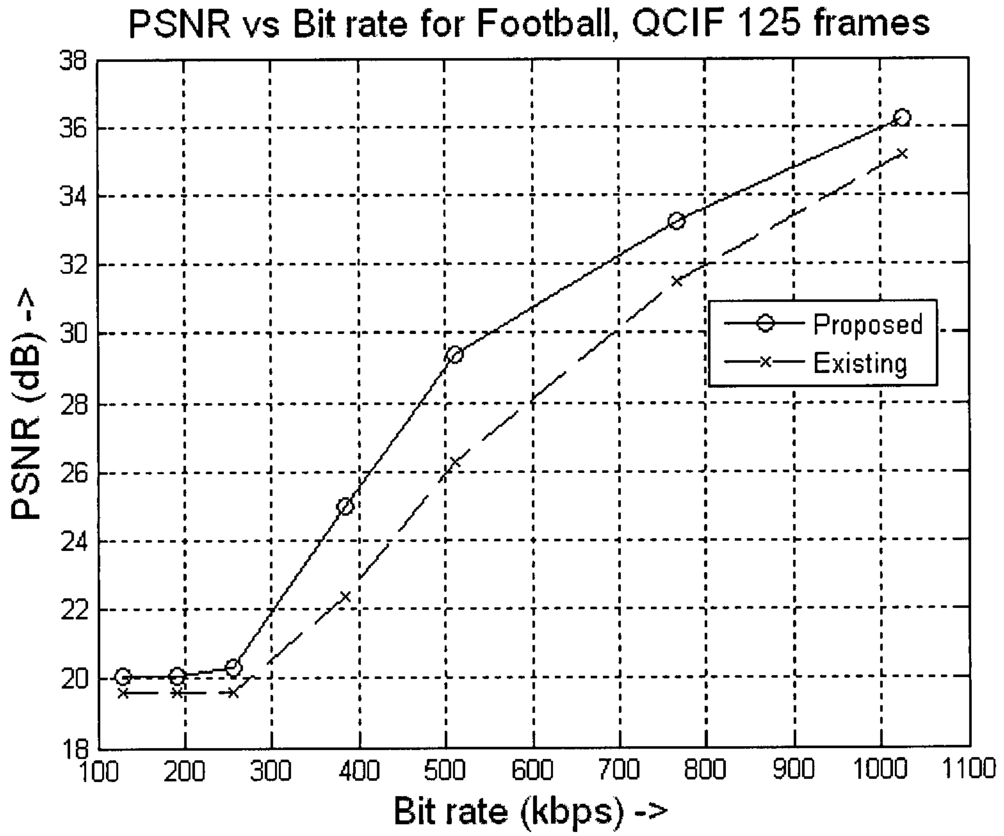


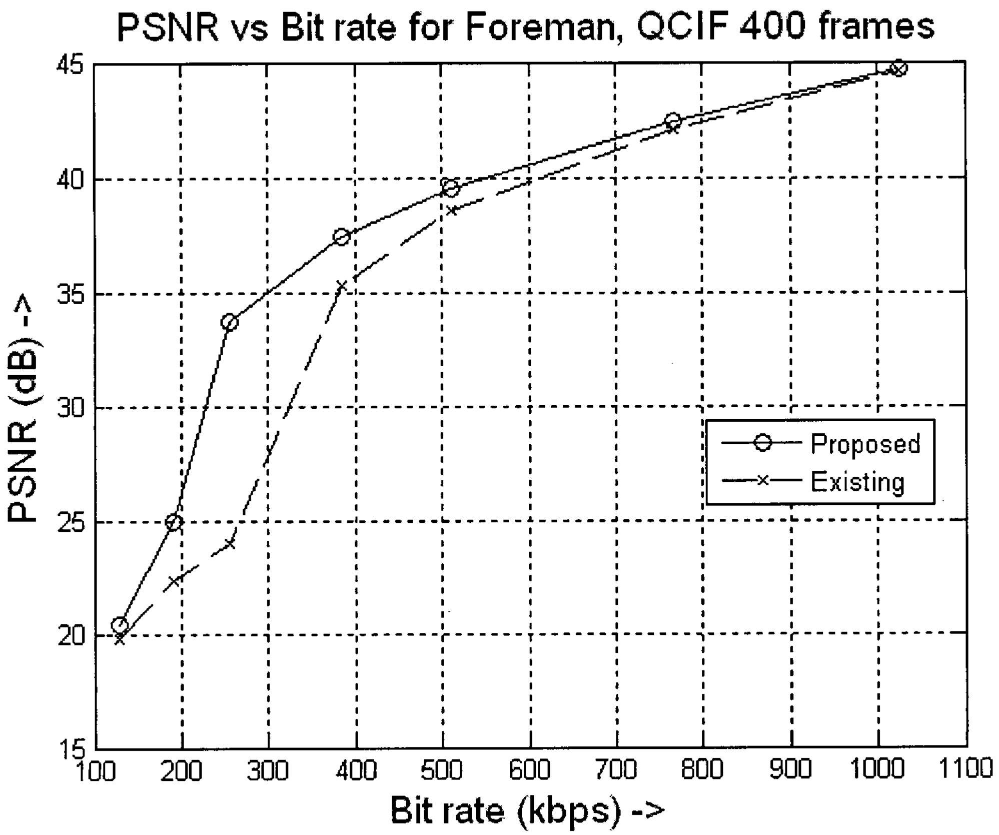
Figure A.2 BER plots for H.264 compression and decompression attack at different bit-rates

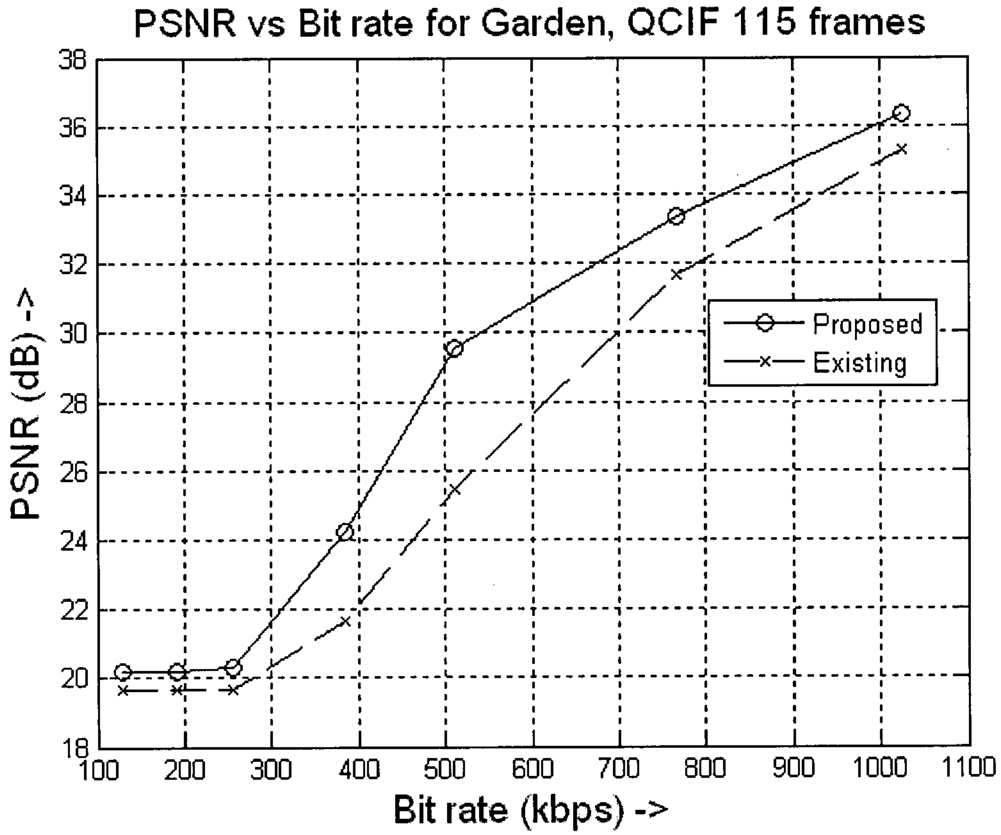
A.3 PSNR plots for H.264 compression and decompression at different bit-rates

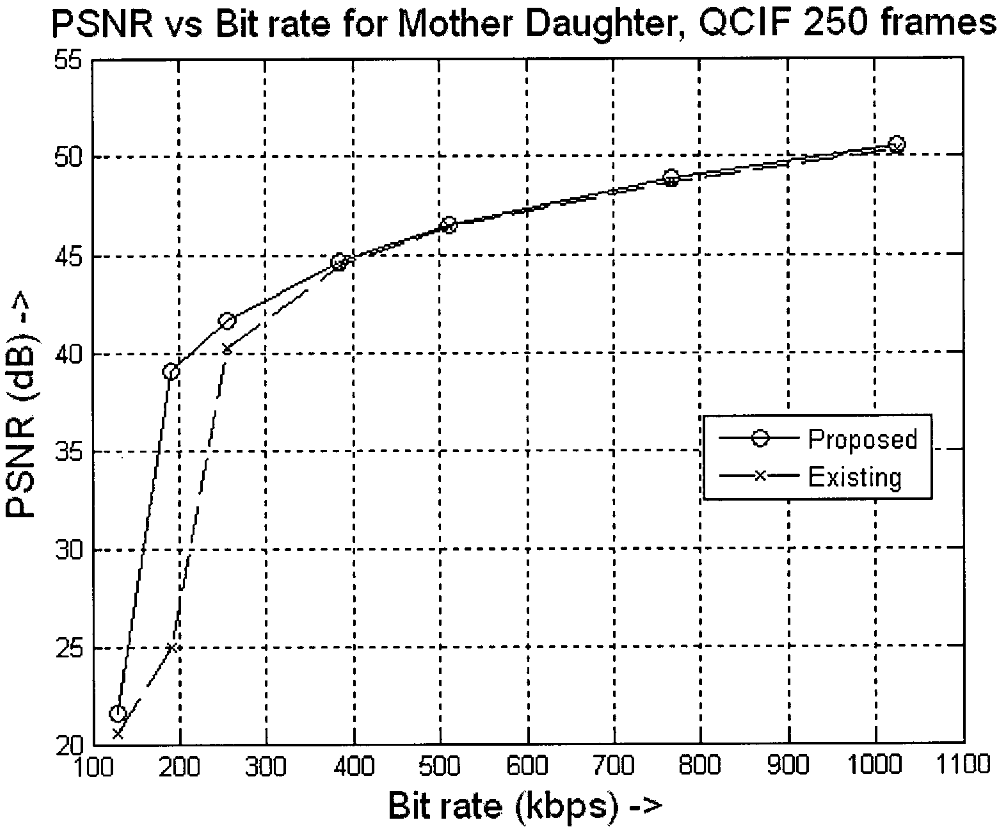


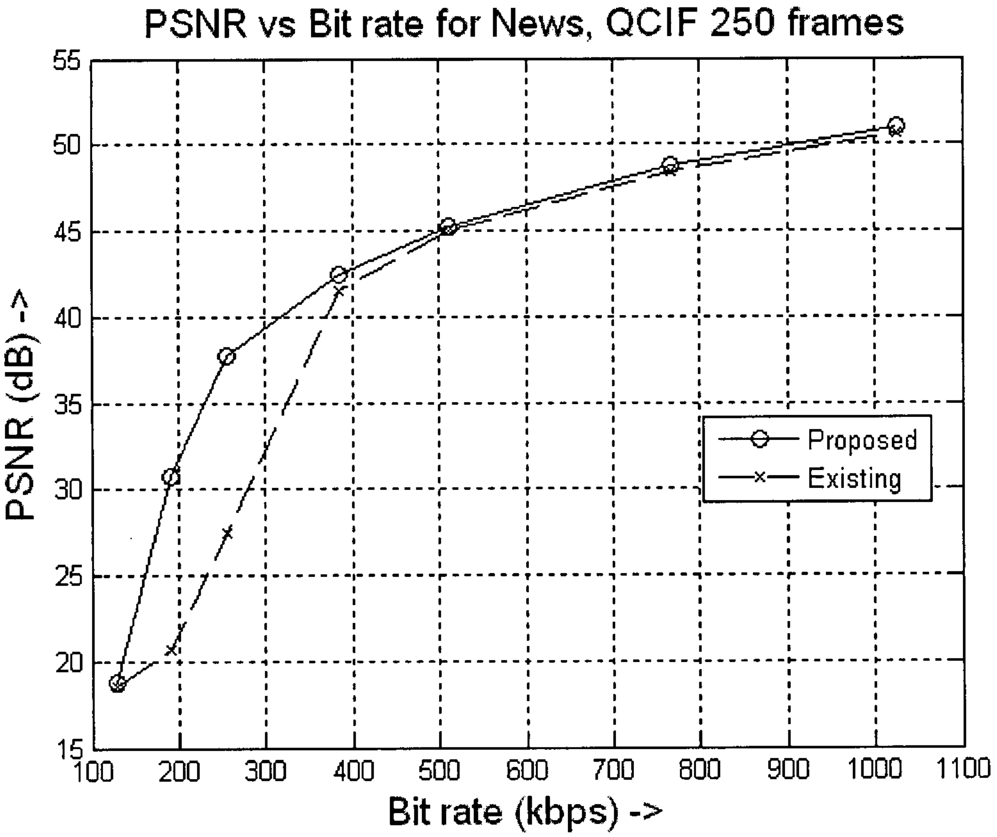

PSNR vs Bit rate for Coastguard, QCIF 250 frames
PSNR (dB) ->
Bit rate (kbps) ->
15
20
25
30
35
40
45
100
200
300
400
500
600
700
800
900
1000
1100
Proposed
Existing

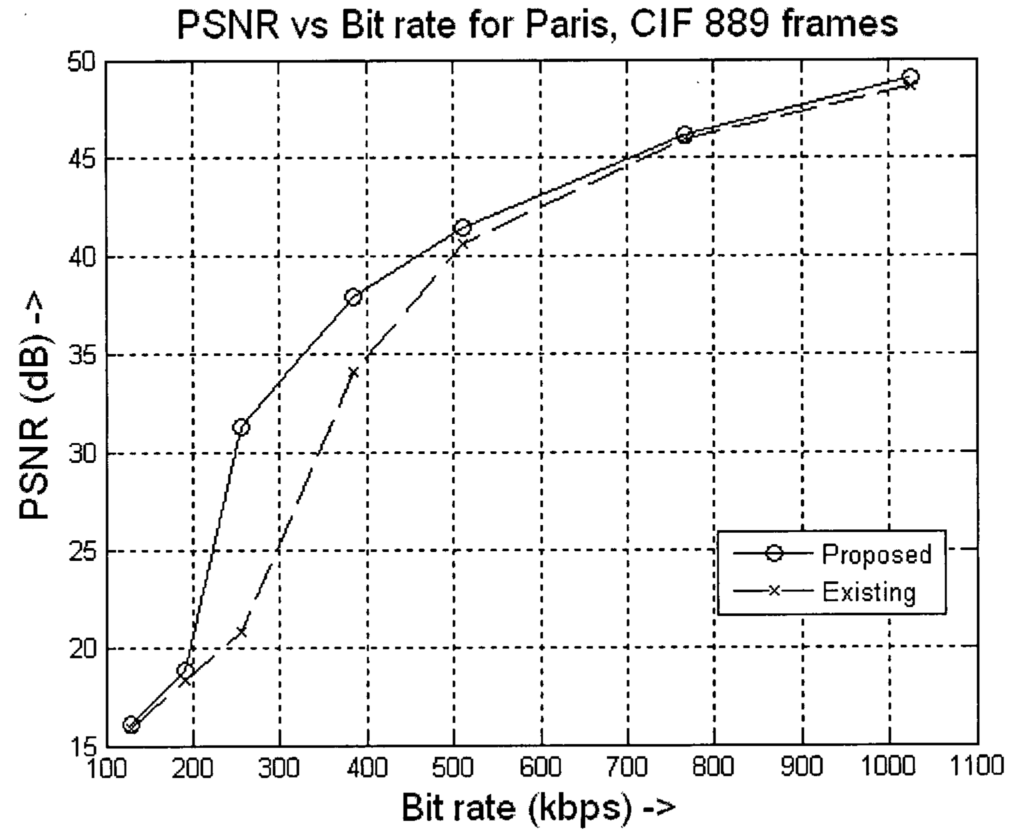

PSNR vs Bit rate for Football, QCIF 125 frames
PSNR (dB) ->
Bit rate (kbps) ->
18
20
22
24
26
28
30
32
34
36
38
100
200
300
400
500
600
700
800
900
1000
1100
Proposed
Existing


PSNR vs Bit rate for Foreman, QCIF 400 frames
PSNR (dB) ->
45
40
35
30
25
20
15
100
200
300
400
500
600
700
800
900
1000
1100
Bit rate (kbps) ->
Proposed
Existing

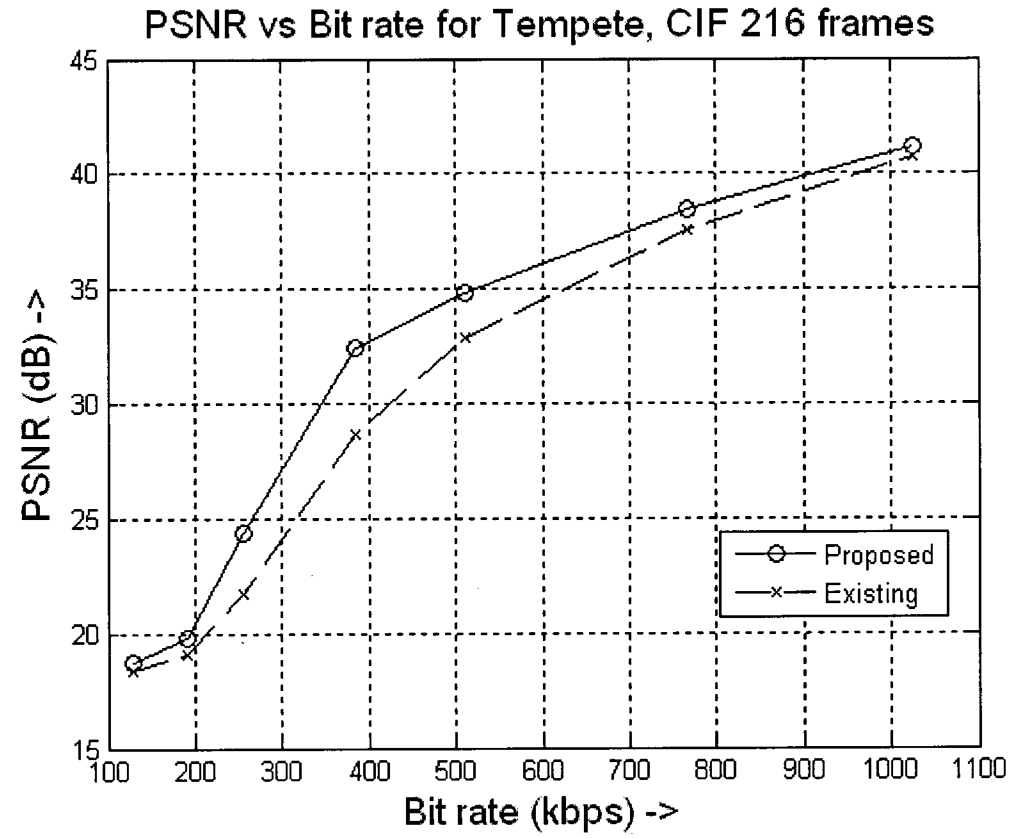

PSNR vs Bit rate for Garden, QCIF 115 frames
PSNR (dB) ->
38
36
34
32
30
28
26
24
22
20
18
100
200
300
400
500
600
700
800
900
1000
1100
Bit rate (kbps) ->
Proposed
Existing


PSNR vs Bit rate for Mother Daughter, QCIF 250 frames
PSNR (dB) ->
Bit rate (kbps) ->
55
50
45
40
35
30
25
20
100
200
300
400
500
600
700
800
900
1000
1100
Proposed
Existing


PSNR vs Bit rate for News, QCIF 250 frames
PSNR (dB) ->
Bit rate (kbps) ->
55
50
45
40
35
30
25
20
15
100
200
300
400
500
600
700
800
900
1000
1100
Proposed
Existing



PSNR vs Bit rate for Paris, CIF 889 frames



PSNR vs Bit rate for Tempete, CIF 216 frames

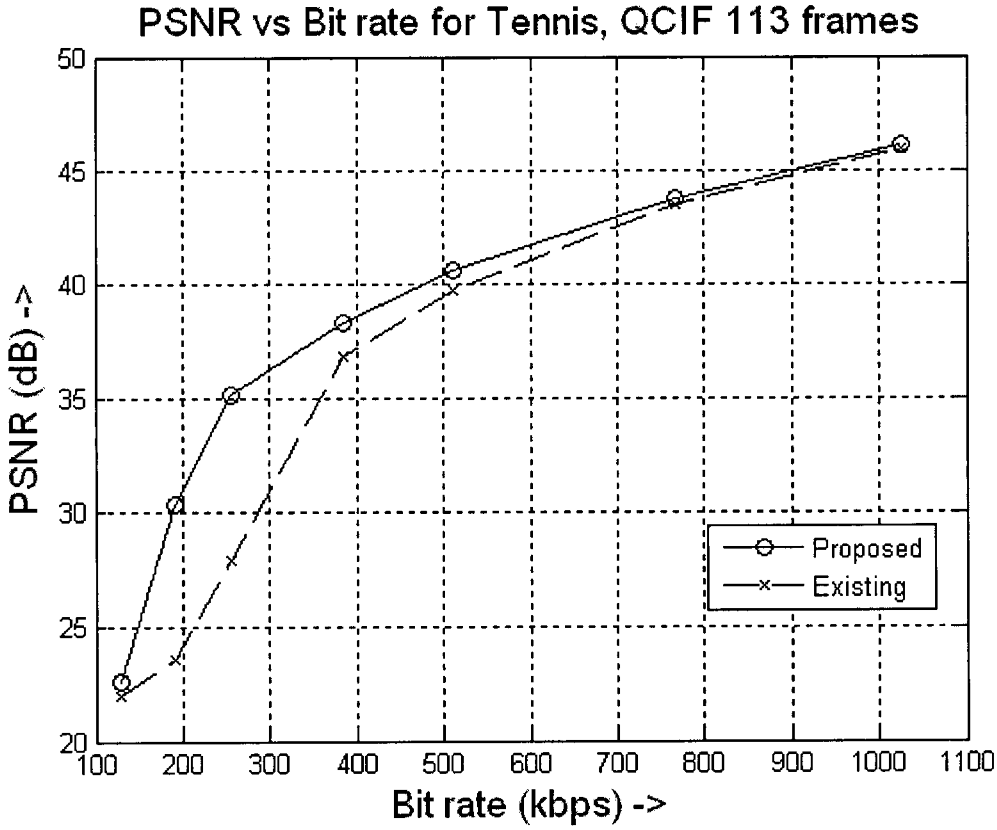
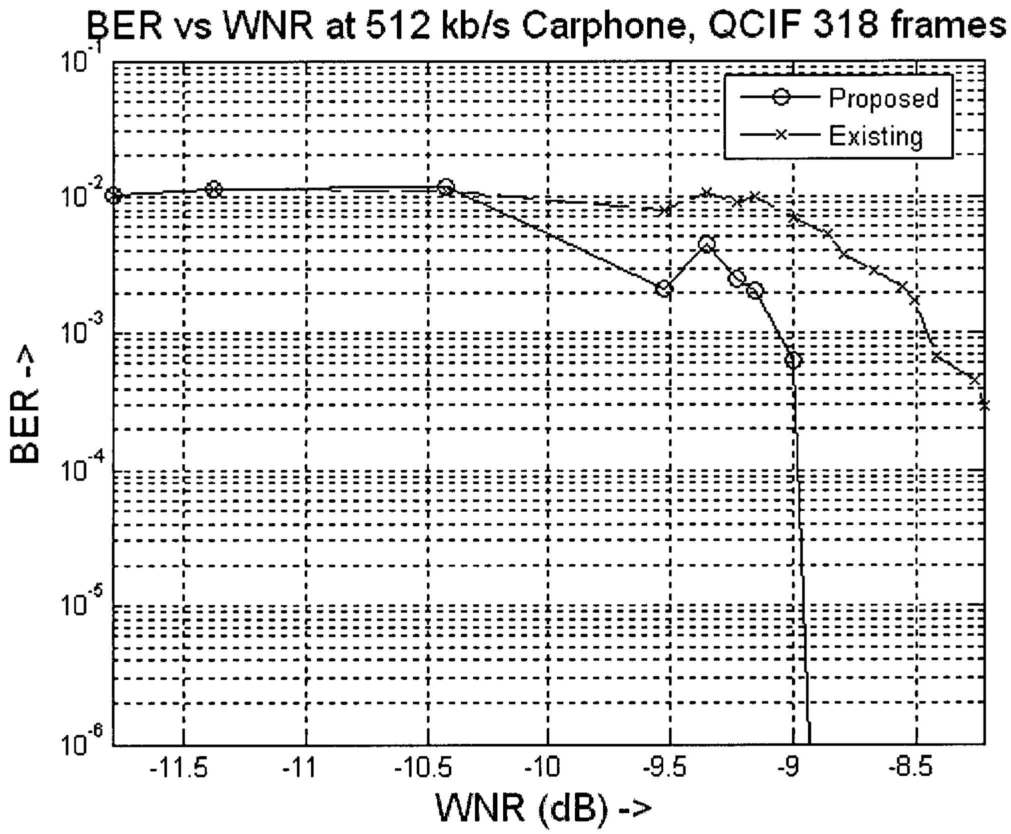
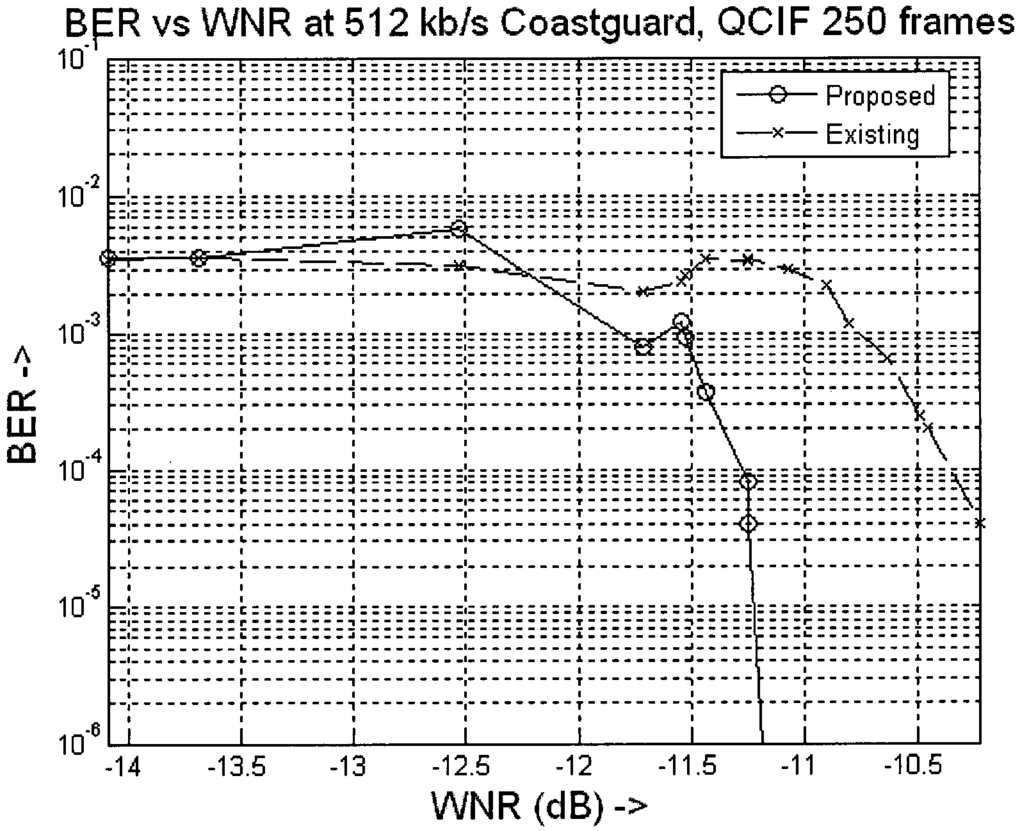
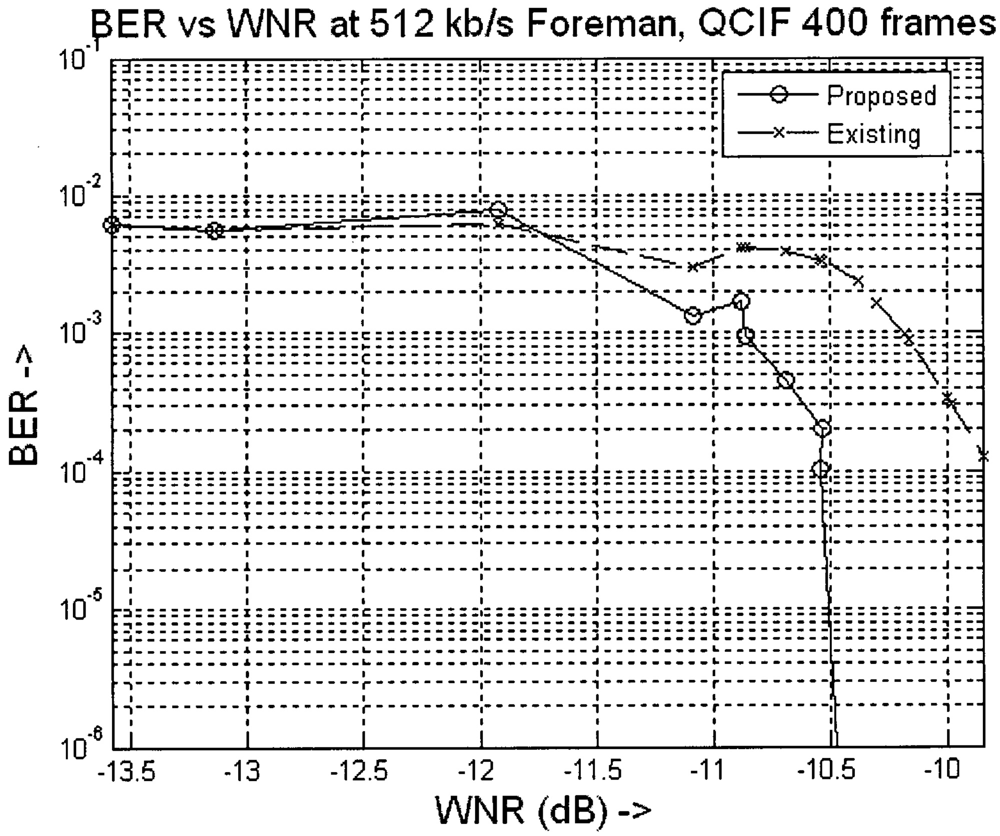


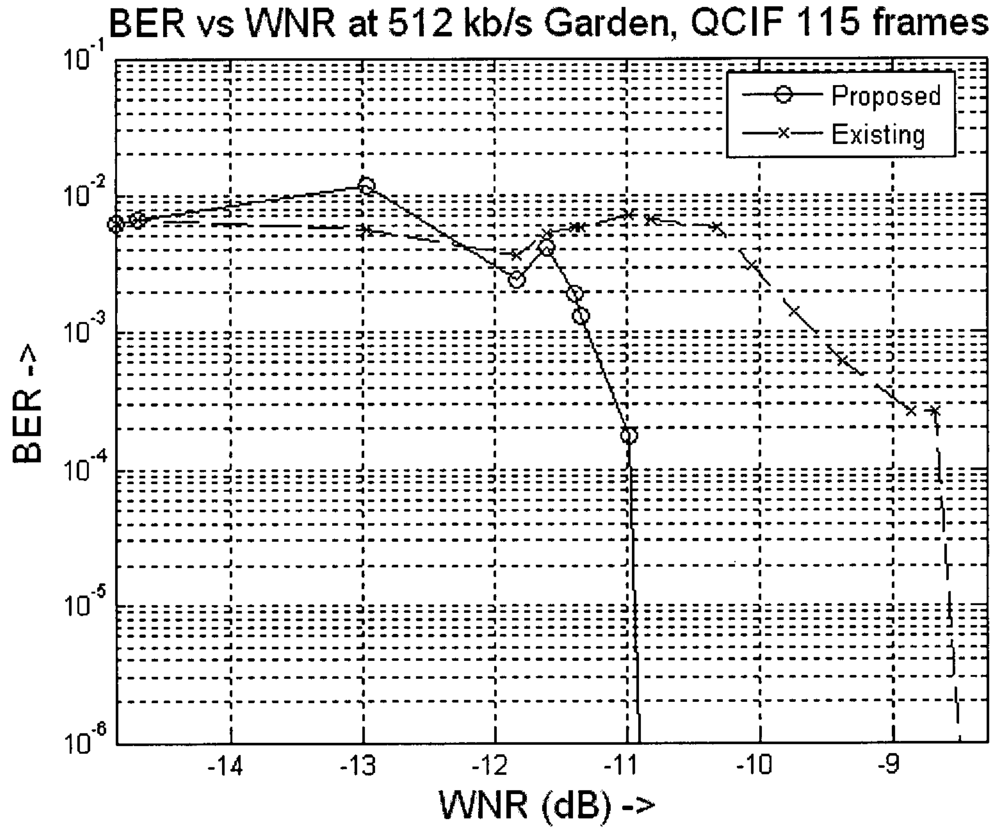
Figure A.3 PSNR plots for H.264 compression and decompression at different bit-rates

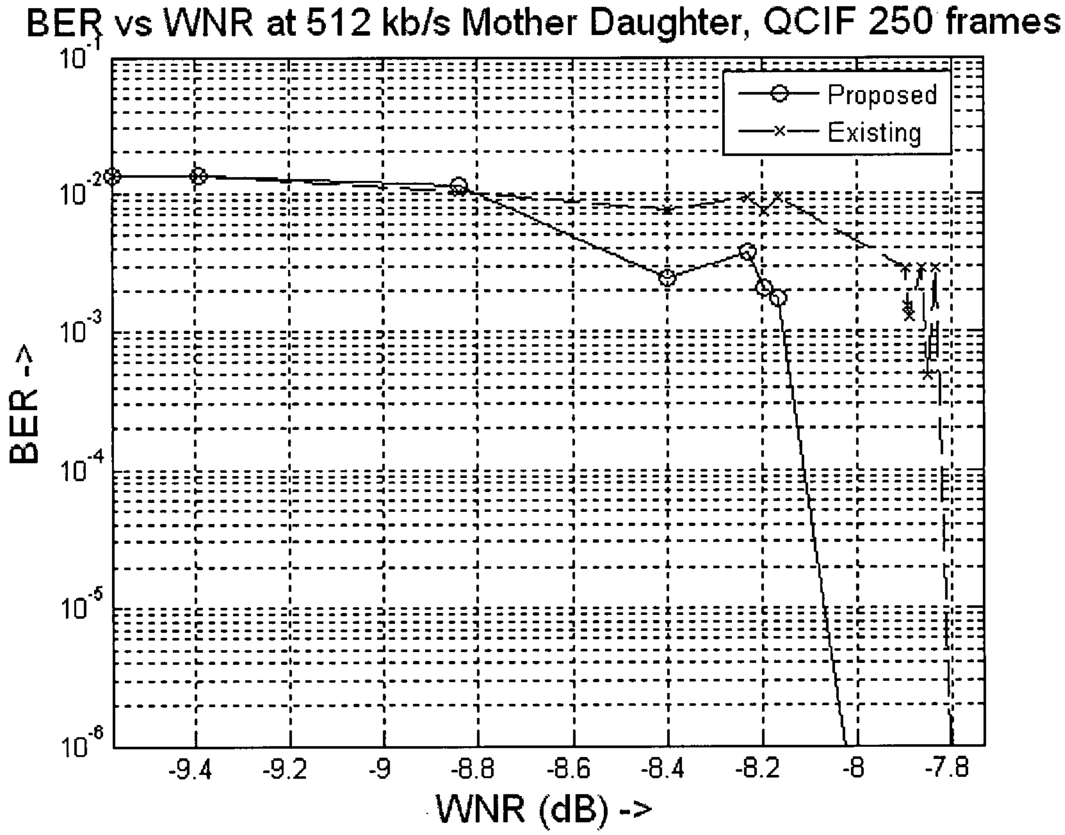
A.4 BER plots for H.264 compression and decompression at 512 kb/s, with different Watermark-to-Noise Ratios (WNRs)




BER vs WNR at 512 kb/s Coastguard, QCIF 250 frames
Proposed
Existing
BER ->
WNR (dB) ->
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
-14
-13.5
-13
-12.5
-12
-11.5
-11
-10.5


BER vs WNR at 512 kb/s Foreman, QCIF 400 frames
Proposed
Existing
BER ->
WNR (dB) ->
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
-13.5
-13
-12.5
-12
-11.5
-11
-10.5
-10


BER vs WNR at 512 kb/s Garden, QCIF 115 frames
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
BER ->
Proposed
Existing
-14
-13
-12
-11
-10
-9
WNR (dB) ->


BER vs WNR at 512 kb/s Mother Daughter, QCIF 250 frames
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
BER ->
Proposed
Existing
-9.4
-9.2
-9
-8.8
-8.6
-8.4
-8.2
-8
-7.8
WNR (dB) ->

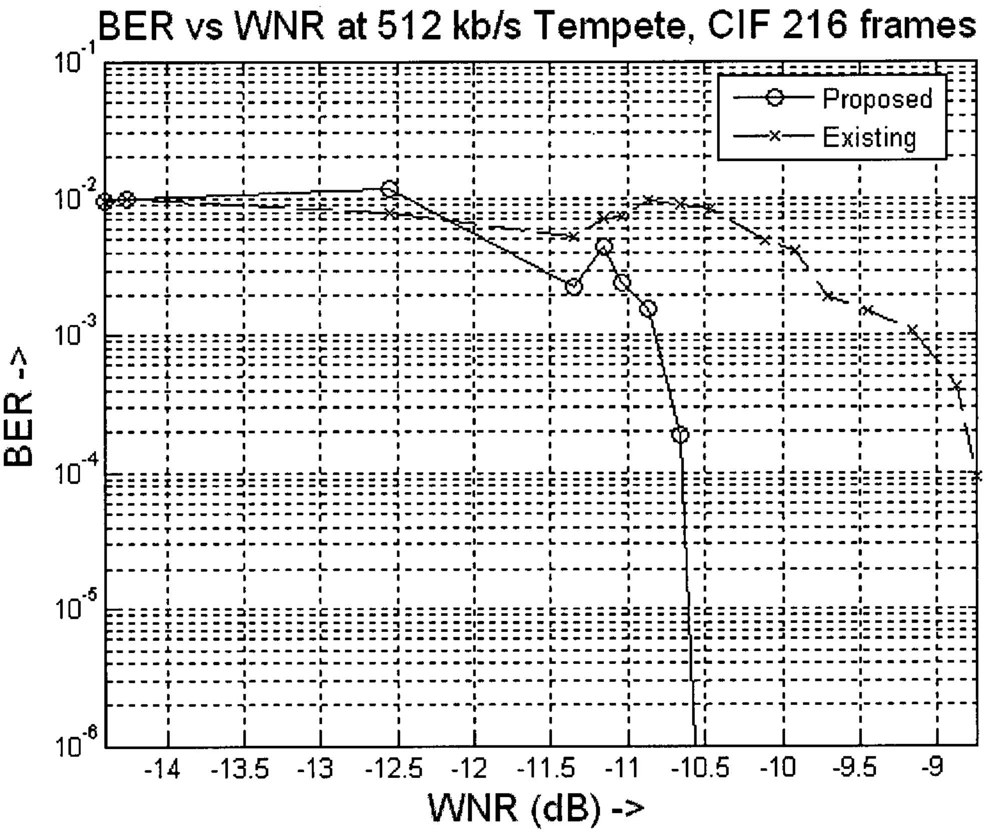
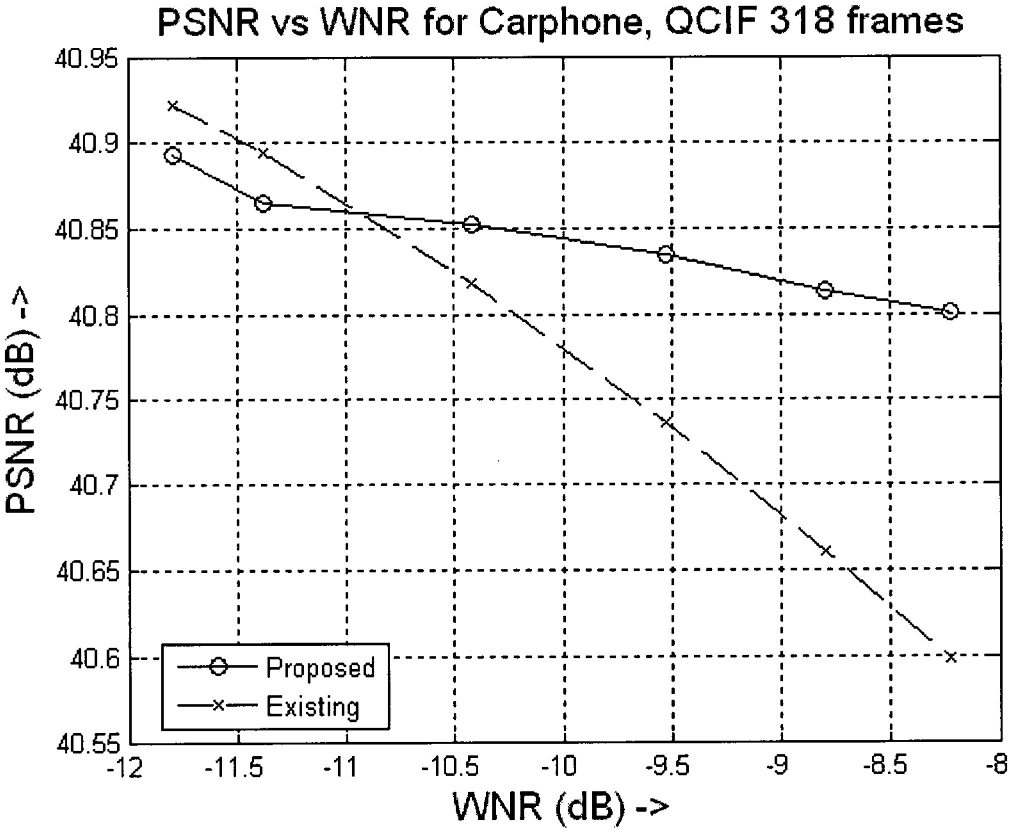
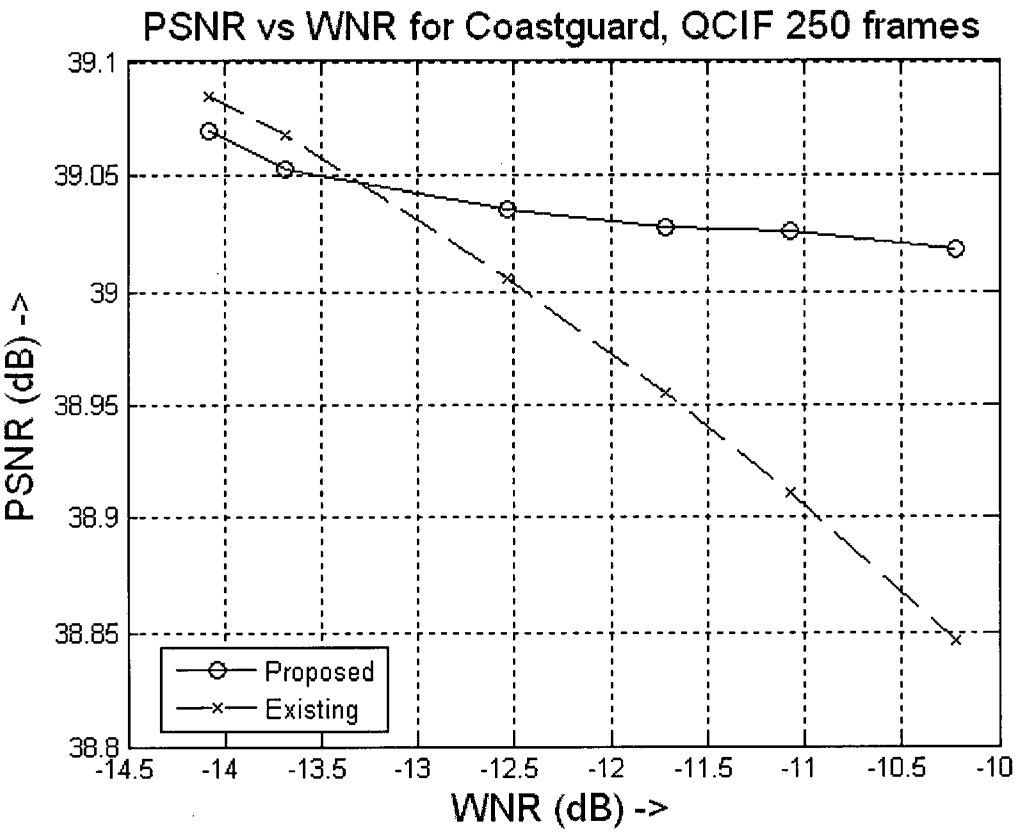
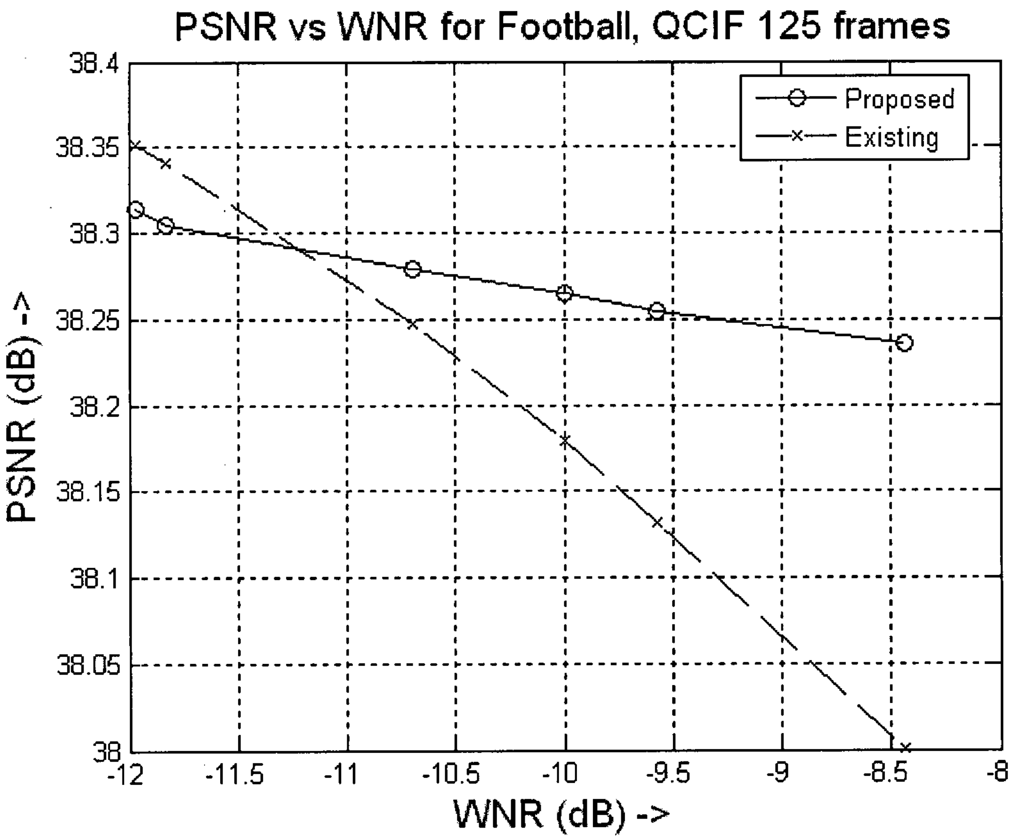


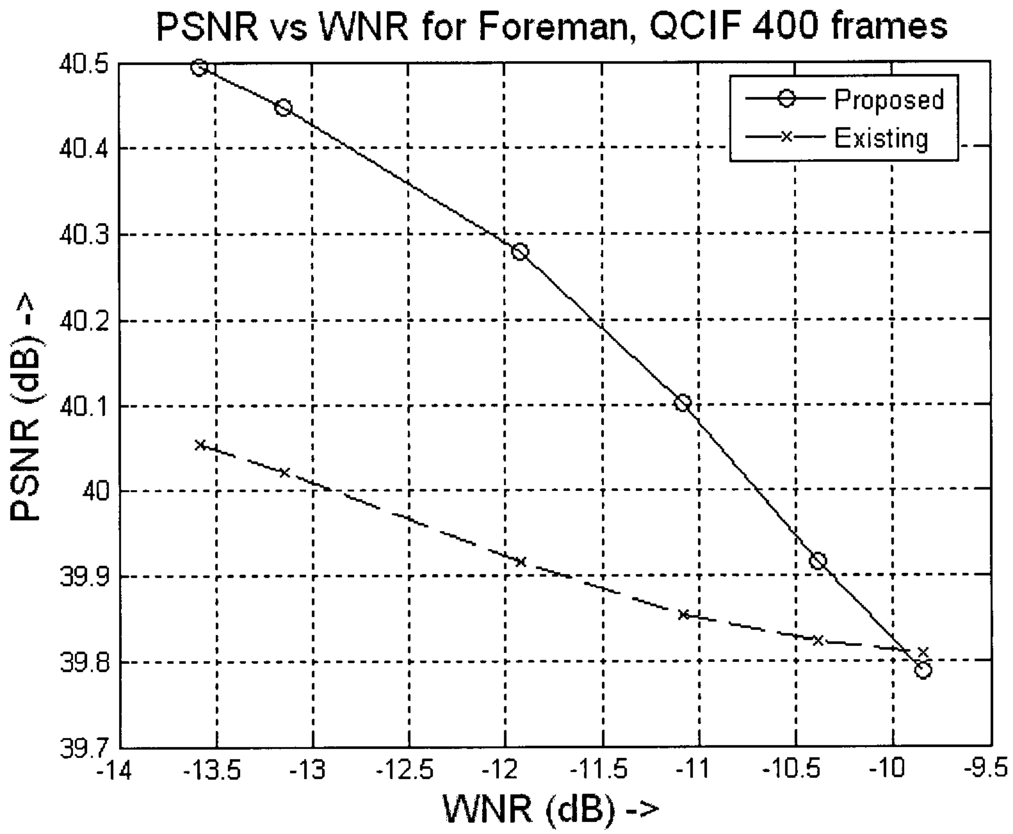
Figure A.4 BER plots for H.264 compression and decompression at 512 kb/s, with different Watermark-to-Noise Ratios (WNRs)

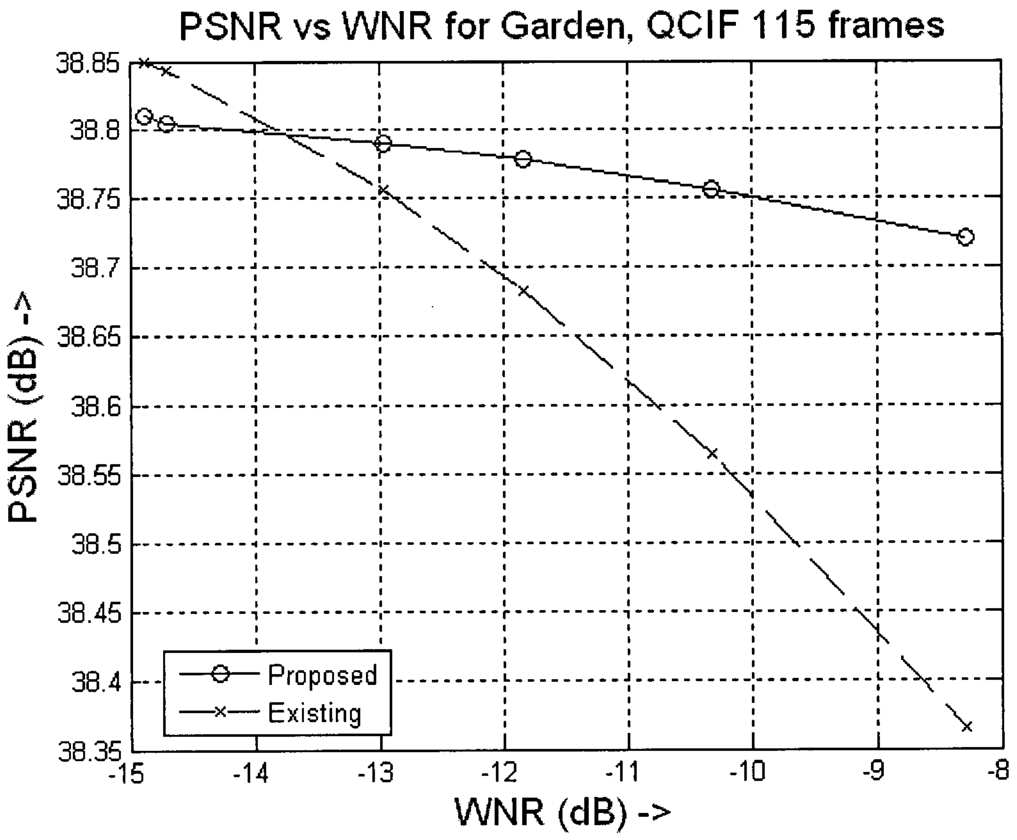
A.5 PSNR plots for H.264 compression and decompression at 512 kb/s, with different Watermark-to-Noise Ratios (WNRs)

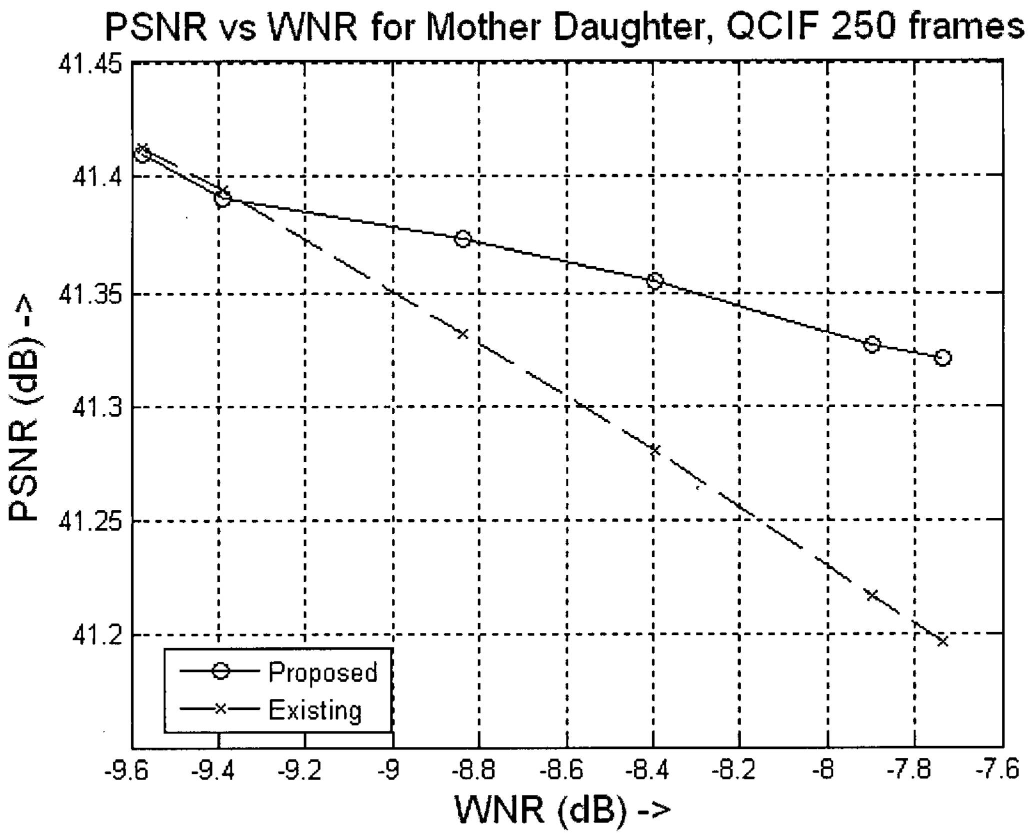


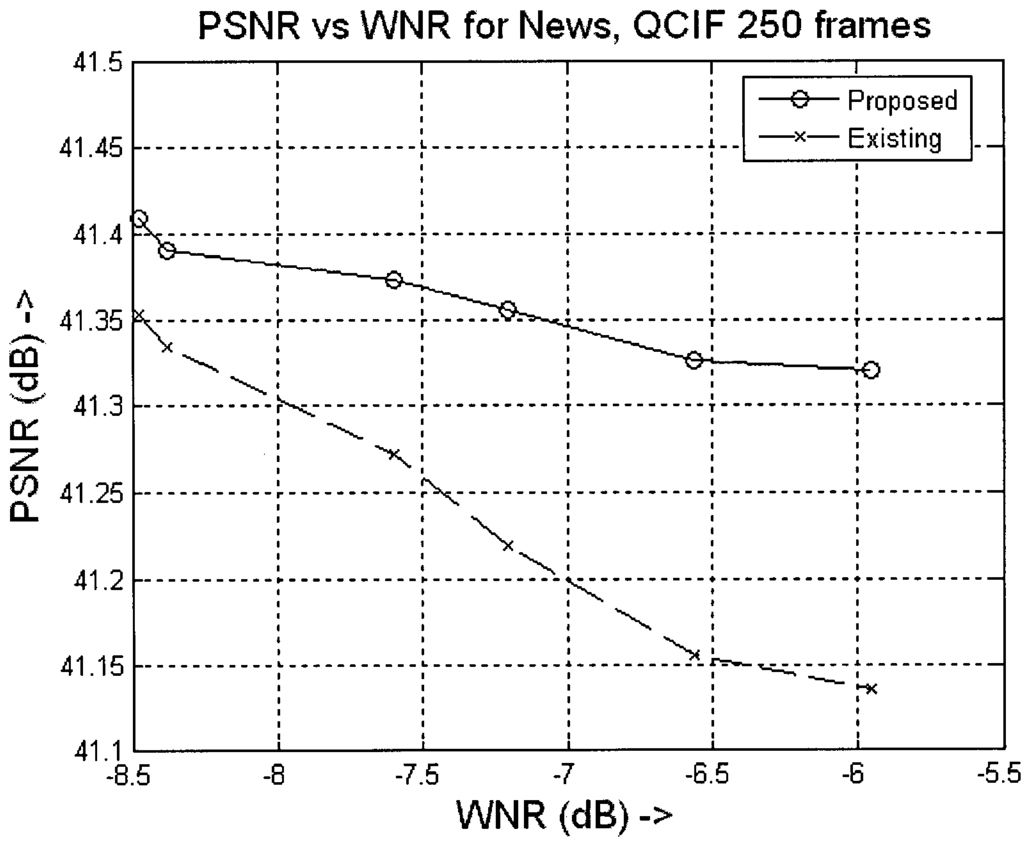

PSNR vs WNR for Coastguard, QCIF 250 frames
PSNR (dB) ->
39.1
39.05
39
38.95
38.9
38.85
38.8
-14.5
-14
-13.5
-13
-12.5
-12
-11.5
-11
-10.5
-10
WNR (dB) ->
Proposed
Existing

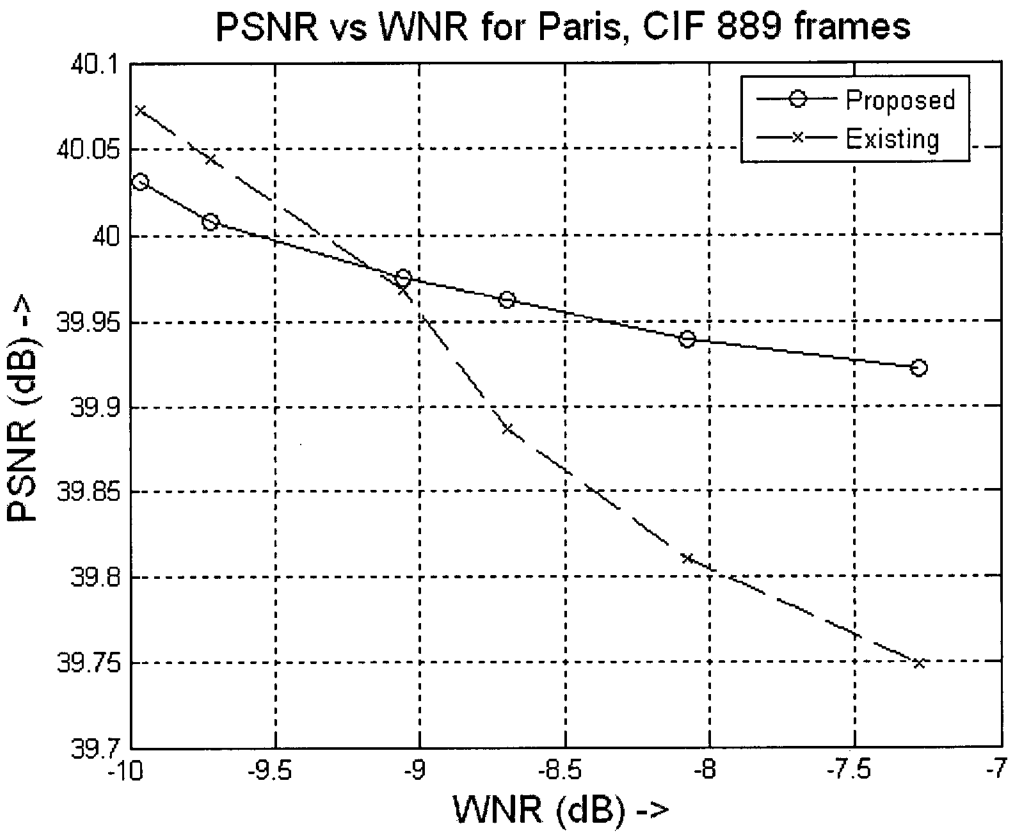

PSNR vs WNR for Football, QCIF 125 frames
PSNR (dB) ->
38.4
38.35
38.3
38.25
38.2
38.15
38.1
38.05
38
-12
-11.5
-11
-10.5
-10
-9.5
-9
-8.5
-8
WNR (dB) ->
Proposed
Existing

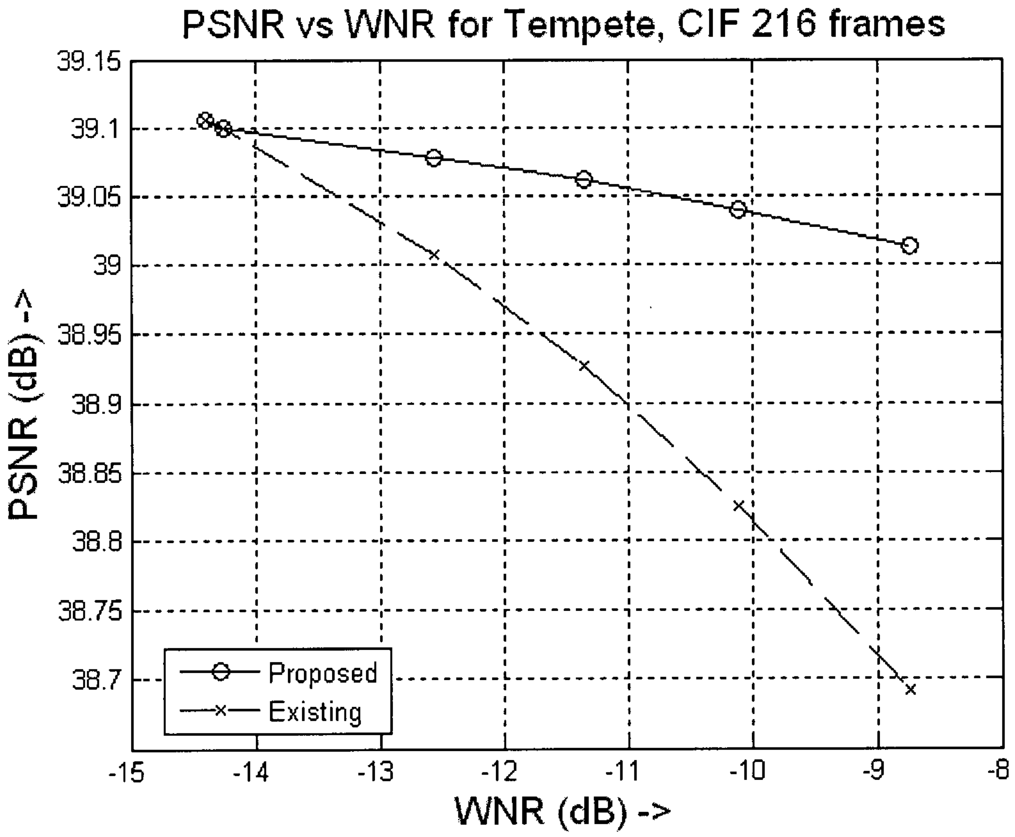

PSNR vs WNR for Foreman, QCIF 400 frames


PSNR vs WNR for Garden, QCIF 115 frames


PSNR vs WNR for Mother Daughter, QCIF 250 frames
PSNR (dB) ->
WNR (dB) ->
41.45
41.4
41.35
41.3
41.25
41.2
-9.6
-9.4
-9.2
-9
-8.8
-8.6
-8.4
-8.2
-8
-7.8
-7.6
Proposed
Existing


PSNR vs WNR for News, QCIF 250 frames
PSNR (dB) ->
WNR (dB) ->
41.5
41.45
41.4
41.35
41.3
41.25
41.2
41.15
41.1
-8.5
-8
-7.5
-7
-6.5
-6
-5.5
Proposed
Existing


PSNR vs WNR for Paris, CIF 889 frames


PSNR vs WNR for Tempete, CIF 216 frames

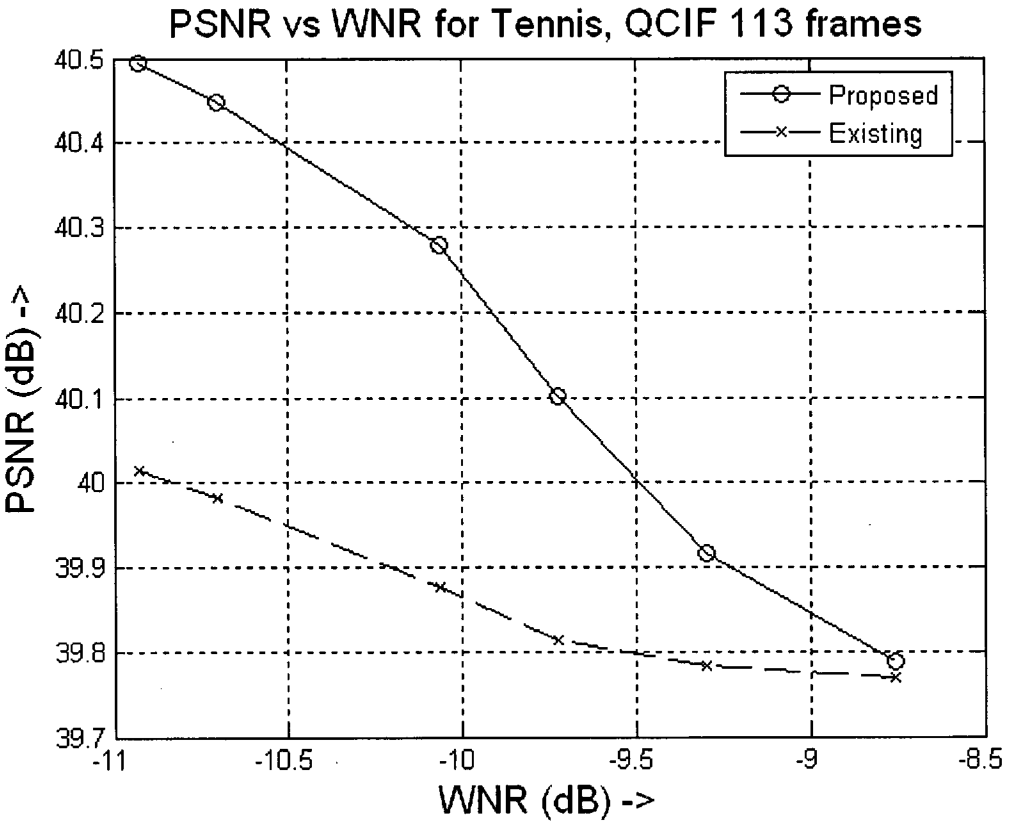


Figure A.5 PSNR plots for H.264 compression and decompression at 512 kb/s, with different Watermark-to-Noise Ratios (WNRs)

Bibliography

[1] I.J. Cox, M.L. Miller and J.A. Bloom, Digital Watermarking, Academic Press, 2002.

[2] G. Langelaar, I. Setyawan, and R. Lagendijk, "Watermarking digital image and video data: A state-of-the-art overview," IEEE Signal Process. Mag., vol. 17, pp. 20--46, Sep. 2000.

[3] Hartung, F., Kutter, M.: Multimedia watermarking techniques. Proceedings of the IEEE, vol. 87, no. 7, pp. 1079-1107, July 1999.

[4] F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video," Signal Process., vol. 66, no. 3, pp. 283--301, May 1998.

[5] J. J. Eggers, R. Buml, R. Tzschoppe, and B. Girod, "Scalar costa scheme for information embedding," IEEE Trans. Signal Process., vol. 15, pp. 1003--1019, Apr. 2003.

[6] M. H. M. Costa, "Writing on dirty paper," IEEE Trans. Info. Thy, vol. 29, no. 3, pp. 439--441, May 1983.

[7] J. J. Eggers and B. Girod, Informed Watermarking. Kluwer Academic Publishers, 2002.

[8] I.J. Cox, M.L. Miller, A.L. McKellips, “Watermarking as communications with side information”, Proc. IEEE, vol. 87, pp. 1127-1141, July 1999.

[9] G. Doërr, J.-L. Dugelay, “A guide tour of video watermarking”, Signal Processing: Image Communication" , 18(4):263-282, 2003.

[10] J.A. Bloom, I.J. Cox, T. Kalker, J.-P. M.G. Linnartz, M.L. Miller, and C.B.S. Traw, "Copy protection for DVD video," Proc. IEEE, vol. 87, no. 7, pp. 1267--1276, July 1999.

[11] D. Boneh and J. Shaw, "Collusion-Secure Fingerprinting for Digital Data," Proc. of CRYPTO '95, Springer-Verlag LNCS, vol. 963, pp. 452--465, 1995.

[12] J. Dittmann, A. Behr, and M. Stabenau, "Combining digital watermarks and collusion secure fingerprints for digital images, " Proc. of SPIE Sec. and Watermarking of Multimedia Cont. I, vol. 3657, no. 13, Jan. 1999.

[13] B. Chen and G. W. Wornell, "Digital watermarking and information embedding using dither modulation," Proc. IEEE Workshop on Multimedia Signal Processing, Redondo Beach, CA, pp. 273--278, Dec. 1998.

[14] B. Chen, "Design and analysis of digital watermarking, information embedding and data hiding systems", Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci. Mass. Inst. Technol., Cambridge, MA, June 2000.

[15] J. Earl, N. Kingsbury, "Spread transform watermarking for video sources", International Conference on Image Processing, 2003.

[16] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), Joint Video Team (JVT), Mar. 2003, Doc. JVT-G050.

[17] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard", IEEE Trans. Circuits Syst. Video Technol., 13:560--576, July 2003.

[18] A. M. Alattar, E. T. Lin, and M. U. Celik, "Digital watermarking of low bit-rate advanced simple profile MPEG-4 compressed video," IEEE Trans. Circuits Syst. Video Technol., vol. 13, pp. 787--800, Aug. 2003.

[19] G. Sullivan and T. Wiegand, "Rate-Distortion Optimization for Video Compression," IEEE Signal Processing Magazine 15, pp. 74--90, Nov. 1998.

[20] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-Constrained Coder Control and Comparison of Video Coding Standards," IEEE Transactions on Circuits and Systems, vol. 13, no. 7, July 2003.

[21] Y.Shoham and A.Gersho, "Efficient Bit Allocation for an Arbitrary Set of Quantizers," IEEE Trans. ASSP, vol.36, pp.1445-1453, September 1988.

[22] T. Wiegand and B. Girod, "Lagrangian multiplier selection in hybrid video coder control," presented at the Proc. ICIP 2001.

[23] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/AVC,"IEEE Trans. Circuits Syst. Video Technol., vol. 13, pp. 598--603, July 2003.

[24] Iain E G Richardson, H.264 and MPEG-4 Video Compression, John Wiley & Sons, September 2003.

[25] R. Wolfgang, C. Podilchuk, and E. Delp, "Perceptual watermarks for digital images and video," Proc. IEEE, vol. 87, no. 7, pp. 1108--1126, Jul. 1999.

[26] C. I. Podilchuk, "Digital image watermarking using visual models," in Proc. Electronic Imaging, vol. 3016, San Jose, CA, Feb. 1996.

[27] A. B. Watson, J. Hu, and J. F. III. McGowan, "Digital video quality metric based on human vision," Journal of Electronic Imaging, vol. 10, no. 1, pp. 20--29, 2001.

[28] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error measurement to structural similarity," IEEE Trans. Image Processing, vol. 13, no. 4, pp.600-612, April 2004.

[29] Karsten Sühring, H.264/AVC Software Coordination, http://bs.hhi.de/~suehring/tml/.

[30] K. Su, D. Kundur, D. Hatzinakos, “Statistical Invisibility for Collusion-Resistant Digital Video Watermarking”, IEEE Trans. Multimedia, Vol. 7, No. 1, Feb. 2005.