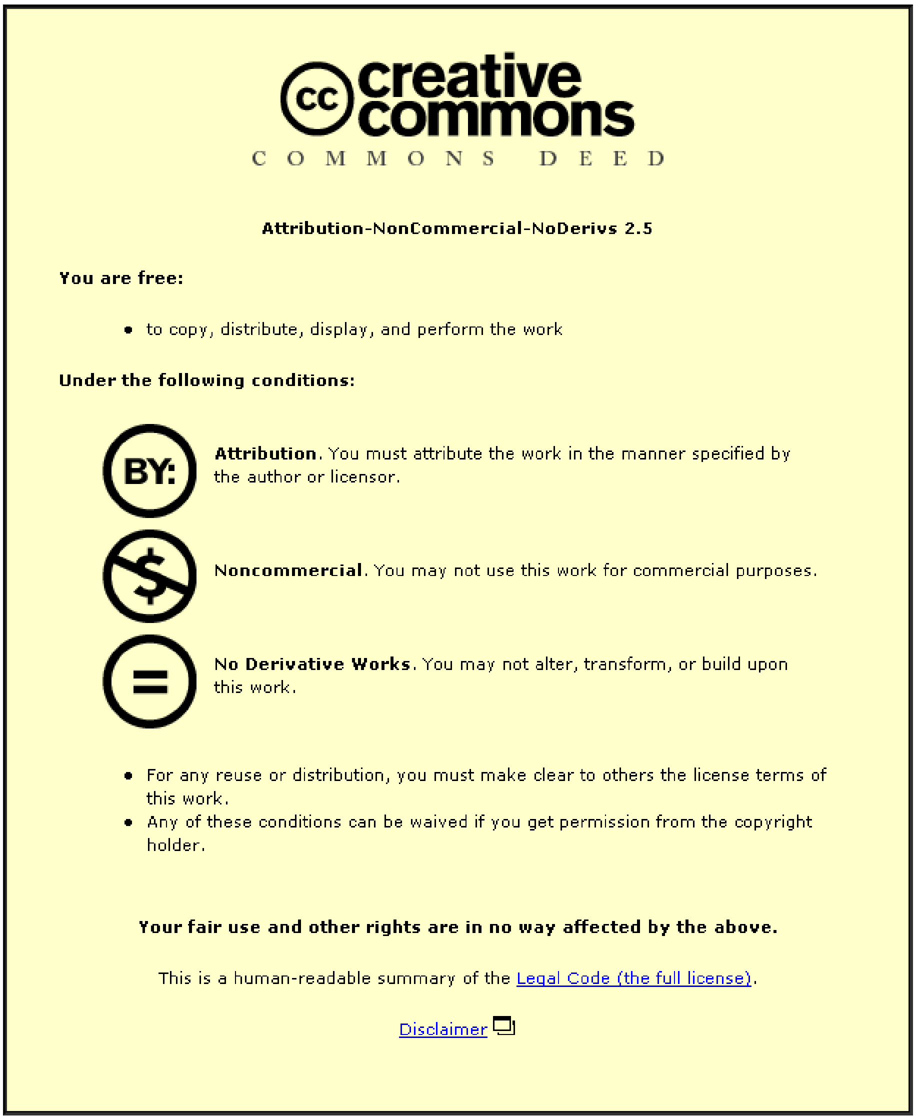


Digital Rights Mangement Techniques for H.264 Video

by

Mohammad Athar Ali

A Doctoral Thesis

Submitted in partial fulfilment
of the requirements for the award of

Doctor of Philosophy
of
Loughborough University

05th August 2011

Loughborough
University

Thesis Access Form

Copy No…………..……………………**Location**………………………………………………….……………...…

Author…………...………………………………………………………………………………………..…….

Title………..

Status of access OPEN / RESTRICTED / CONFIDENTIAL

Moratorium Period:…………………………………years, ending…………../…………200……………………….

Conditions of access approved by (CAPITALS):…………………………………………………………………

Supervisor (Signature)…………………………………………...…………………………………...

Department of……………………………………………………………………...…………………………………

Author's Declaration: *I agree the following conditions:*

Open access work shall be made available (in the University and externally) and reproduced as necessary at the discretion of the University Librarian or Head of Department. It may also be digitised by the British Library and made freely available on the Internet to registered users of the EThOS service subject to the EThOS supply agreements.

*The statement itself shall apply to **ALL** copies including electronic copies:*

This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

Restricted/confidential work: All access and any photocopying shall be strictly subject to written permission from the University Head of Department and any external sponsor, if any.

Author's signature………………………………………Date…………………………………...…………...……...

<table>
<tr><td colspan="4"><b>users declaration:</b> for signature during any Moratorium period (Not Open work):
<b><i>I undertake to uphold the above conditions:</i></b></td></tr>
<tr><td>Date</td><td>Name (CAPITALS)</td><td>Signature</td><td>Address</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>

Research Student Office, Academic Registry
Loughborough University, Leicestershire, LE11 3TU, UK
Switchboard: +44 (0)1509 263171 Fax: +44 (0)1509 223938

Certificate of Originality

This is to certify that I am responsible for the work submitted in this thesis, that the original work is my own except as specified in acknowledgements or in footnotes, and that neither the thesis nor the original work contained therein has been submitted to this or any other institution for a higher degree.

. . . .

Mohammad Athar Ali

05th August 2011

Dedicated to Abbu, Ammi,
Parveen, Maaz and Faiza

Abstract

This work aims to present a number of low-complexity digital rights management (DRM) methodologies for the H.264 standard. Initially, requirements to enforce DRM are analyzed and understood. Based on these requirements, a framework is constructed which puts forth different possibilities that can be explored to satisfy the objective. To implement computationally efficient DRM methods, watermarking and content based copy detection are then chosen as the preferred methodologies.

The first approach is based on robust watermarking which modifies the DC residuals of 4×4 macroblocks within I-frames. Robust watermarks are appropriate for content protection and proving ownership. Experimental results show that the technique exhibits encouraging rate-distortion (R-D) characteristics while at the same time being computationally efficient.

The problem of content authentication is addressed with the help of two methodologies: irreversible and reversible watermarks. The first approach utilizes the highest frequency coefficient within 4×4 blocks of the I-frames after CAVLC entropy encoding to embed a watermark. The technique was found to be very effective in detecting tampering. The second approach applies the difference expansion (DE) method on IPCM macroblocks within P-frames to embed a high-capacity reversible watermark. Experiments prove the technique to be not only fragile and reversible but also exhibiting minimal variation in its R-D characteristics.

The final methodology adopted to enforce DRM for H.264 video is based on the concept of signature generation and matching. Specific types of macroblocks within each predefined region of an I-, B- and P-frame are counted at regular intervals in a video clip and an ordinal matrix is constructed based on their count. The matrix is considered to be the signature of that video clip and is matched with longer video sequences to detect copies within them. Simulation results show that the matching methodology is capable of not only detecting copies but also its location within a longer video sequence. Performance analysis depict acceptable false positive and false negative rates and encouraging receiver operating characteristics. Finally, the time taken to match and locate copies is significantly low which makes it ideal for use in broadcast and streaming applications.

Acknowledgements

This journey of earning a PhD has not been without its share of ups and downs but it was one person who was instrumental in seeing me through it; my supervisor, Dr. Eran Edirisinghe. I would like to express my deepest gratitude and heartfelt thanks to Dr. Edirsinghe for making this journey more enjoyable than I could possibly imagine. He has been a beacon of inspiration and a source of motivation all through. Every meeting with him was a confidence booster for me. I am amazed by his personality and his caring attitude, not just towards me but towards all his students. I am confident that he is the best PhD supervisor that any student can ever hope to have. Hats off to you Sir!

Further thanks go out to members of the Digital Imaging Research Group (DIRG) for putting the fun into research and giving me a good reason to come to the lab. I am sure it would not have been possible for me finish this work in time, had it not been for the support and the encouragement from this group.

I would also like express my gratitude to Mr. G. S. Samra, Mr. S. Bedi and Mr. K Sahnsi for providing the IT support I needed to keep my computer systems running smoothly. I am sure I have troubled them more than I should have, even though they never complained. Without them it would not have been possible for me to finish my experiments within the scheduled time.

There are a few other people who I would like to mention individually without whom this journey would have not only been difficult but very boring indeed. I am forever indebted to Afizan Azman, Andrew J. Leonce, Iffat Zafar, Muhammad Akramshah Ismail, Nesreen Otoum, Sara Saravi and Usman Zakir for being an extended family to me. They were there whenever I needed to talk, either about my research or any other subject, not to forget the regular trips and get-togethers they organized. I also highly appreciate the tips and the feedback I received from them during the course of my research. Sincere thanks are also extended towards my neighbour in the lab, Mr. Sazalinsyah Razali for giving me information and tips on many aspects ranging from writing scripts for plotting graphs to easily transferring my files across different computers. He has an amazing repository of knowledge when it comes to software and I am lucky to have picked up a few tips from him which I am sure will help me a lifetime.

I would also like to extend my sincere thanks to my peers, staff and faculty at the Department of Computer Science at Loughborough University. In particular, my special thanks go out to the Head of the Department, Dr. I. Philips who went out of his way to make it possible for me to travel to South Korea to attend a conference. A very big thanks also goes out to Ms. J. Poulton, Ms. Christine Bagley, Ms. J. McOuat and Mr. P. Holligan. They never cease to amaze me with the way they make so many things possible. My appreciation also goes out to Dr. Qinggang Meng for reviewing my first and second year progress review reports and providing invaluable suggestions and feeback which helped me immensely.

I would also like to express my sincere gratitude towards my employer, the Department of Computer Engineering at Aligarh Muslim University, Aligarh, India for granting me study leave to pursue PhD and Loughborough University for sponsoring my studies.

Last, but not the least, my love, gratitude and sincere appreciation go towards my family in India, who have been my pillar of strength, confidence and hope during the long lonely months I spent away from them. They never gave up hope on me and would talk to me on a daily basis. They would never let an iota of doubt creep into my confidence and were the driving force into keeping my focus and concentrating on my goals. Without them, surely, this work would not have been possible.

Mohammad Athar Ali

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The rapid growth of broadband Internet has led to the easy exchange digital multimedia information. It takes just a few minutes or sometimes even seconds to transfer digital multimedia data from one part of the globe to the other. In addition, the easy availability of powerful computing resources and multimedia software has made processing and editing of videos a very easy task. A person does not need any special knowledge to process, edit and manipulate videos. This means that digital video can be easily copied, manipulated and retransmitted. As a result, copyrights could be violated and is a major issue with content production companies. An analysis by LEK for Motion Picture Association in 2005 [1] estimated that major motion picture studios lost $6.1 billion due to movie piracy in 2005 and out this $2.3 billion was lost due to internet piracy. Of course, it is safe to assume that this figure must be significantly higher at present. As a result, content creators and providers are always searching for more secure methods to distribute of their content online. The problem involves three factors: (1) content protection and proof of ownership (2) content authentication and (3) copy detection. Content protection implies that certain methodologies should be in place that prevents any unauthorized copying of the content; while proving content ownership means a technique which identifies the rightful owner of the content. Content authentication has its significance in situations where it may be important to verify that the content has not been edited, damaged or altered over a period of time by an unauthorized user. Copy detection involves the use of methodologies to detect the presence of a modified copy of an original video within a larger database, within a broadcast or within a longer video sequence. For ease of discussion, from this point on for the remaining part of this chapter, the term 'protection' would be used to include all the above factors.

The severity of the issue has generated considerable interest in the research community. Many different approaches have been suggested that address either one or all of the factors mentioned above. Most of the approaches have been derived from the techniques that were used to protect digital content such as images and voice. But recently some techniques have been developed specifically for video. This is so because video content has certain characteristics that set it apart from other types of digital content. Unfortunately, it is these very characteristics that make the development of protection mechanisms for video content that much more difficult. For instance, video by nature has substantial redundant information and even a part of this information can be used by pirates to create satisfactory copies which bypass all copy detection mechanisms. In fact, newer and more powerful techniques are emerging on a day-to-day basis that makes illegal activities on videos a very mundane task.

1.1 Motivation

With piracy on the rise and illegal duplication of video content becoming very common, content creators and owners are facing mounting losses. Digital data pirates always seem to be a step ahead of even the latest and the most sophisticated content protection techniques. Newer and stronger attacks are being launched that render the protection mechanism useless. Content owners/creators not only need a system that repels most of the attacks but also detects them; while at the same time preserving the characteristics of the video.

The H.264 is the latest video standard rapidly being adopted for a number of applications. Increasingly large amounts of content is now being encoded and distributed under this standard. Correspondingly, more and more H.264-based content is being illegally modified and copied. In addition, the standard is still evolving with newer features being added to it. This, in turn, means that newer, better and more secure protection algorithms need to be developed as the standard evolves. The high compression ratio offered by the standard means that the encoded video is suitable for transmission even on low-bitrate channels. This makes the task of developing a protection mechanism that much more difficult since any modification to the encoder may result in an increase of the bitrate by a significant amount thus defeating the very purpose of compression. A protection system that guarantees only a minimal increase in bitrate is highly desirable. This work would aim to address these requirements by analyzing those aspects of the H.264 codec that can be utilized to design an efficient Digital Rights Management (DRM) system. More specifically, the focus would be on maintaining an optimum rate-distortion characteristic. This implies that a certain level of quality for the video content would be maintained for a given bitrate even with the protection mechanism in place.

Digital pirates could perform a variety of attacks on video data. That includes making illegal copies, modifying the video content, claiming ownership etc. Correspondingly, different methods have been developed to repel/detect these attacks. Encryption, cryptographic hash algorithms and robust watermarks are used to prevent video data from being copied illegally and to authenticate the content as well as a legitimate user. Fragile, hybrid watermarks, content-based copy detection, on the other hand, are used for source/content authentication as well as to detect modified copies of an original video data. However, most of the aforementioned techniques lead to an unacceptable processing and transmission overhead when it comes to the H.264 standard. Thus the motivation behind this work was to develop content protection mechanisms that do not compromise on the high performance characteristics of the H.264 standard while at the same time offering effective protection.

1.2 Research Aims and Objectives

As will be explained in Chapter 3, content protection, proof of ownership, content authentication and copy detection are all aspects of a collective term known as Digital Rights Management. However, DRM techniques based on encryption and cryptographic hash algorithms are usually computationally complex and lead to a significant computational overhead. This in unacceptable since the H.264 encoder is in itself very complex and has a significant encoding time of its own. Burdening the encoder by incorporating an encryption or a cryptographic algorithm would significantly limit its application especially in situations where the bandwidth is constrained and the receiving end devices are low-power portable devices.

Thus, within this study, watermarking and content-based copy detection (CBCD) are explored as other possible viable alternatives to enforce DRM. The justification for choosing these two approaches is the simple fact that not only are they computationally efficient but also effective in enforcing both, content protection and content authentication. But both aspects have conflicting requirements and most of the algorithms proposed address either of the two. When it comes to watermarking, an effective algorithm should be transparent, secure, computationally efficient, unambiguous, and readily extractable. Transparency implies that the effect of the watermark on the host data should be negligible, at least perceptually. Security implies that the strength of the watermarking system should rely on the use of secret keys rather than obscuring the watermarking algorithm. A good watermarking algorithm should not place a huge burden on computational resources when it is executed. Unambiguity is a feature that allows unique identification of the data owner. Easy extractability would permit the content owner/creator or a legitimate user to easily extract the watermark.

This purpose of this study is to develop watermarking techniques for DRM that satisfy most of the requirements mentioned above. In addition, the performance of the watermarking algorithms developed are evaluated for their effectiveness by simulating a variety of attacks on the watermark. Their performance is also compared against other existing algorithms. The comparisons are made in terms of the degradation in the quality of the video after the watermark is embedded, and the resulting increase in the bitrate. Again these are contradictory factors and a balance has to be found that provides the best video quality at any given bitrate.

CBCD methods consider certain features within the video itself as a “watermark” in order to detect copies. In line with developing computationally efficient DRM methods, the aim is to look for such features within a video sample that can be extracted efficiently and used as signatures to detect copies. The performance of the developed algorithm is checked under a variety of conditions such as searching for copied video clips within a large video database or within a long video sequence. Again the feasibility of the technique developed would depend on performing the search and matching within a reasonable amount of time.

1.3 Problem Overview

A number of DRM algorithms have been proposed for video in general and H.264 in particular. However, most of these methods fail to take into account the novel features incorporated within the H.264 standard. As a result, the techniques reported may be quite efficient and effective for earlier video standards but fail to maintain the same level of performance when applied to H.264 video. Therefore, recently, a significant amount of research has been carried out which focuses on developing DRM systems custom-made for this standard. This work also follows the argument that to ensure that the developed DRM methodology does not have an adverse effect on the performance of the H.264 codec, its functioning must be thoroughly understood. To summarize, within this study, the problem of designing computationally efficient DRM systems for the H.264 standard follows these steps:

1. Study the functioning of a typical DRM system and understand its requirements.
2. Understand and identify aspects of the H.264 standard that can be exploited to design computationally efficient DRM systems.
3. Design computationally efficient DRM enforcement methods based on the aspects identified in step 2.
4. Evaluate the performance of the algorithms designed in step 3 and perform a comparative evaluation with other similar reported methods.

1.4 Scholarly Contribution

The proposed work thus divides the problem outlined in the previous section into three broad categories of content protection/proof of ownership, content authentication and copy detection. For each of these categories a different DRM method is developed, simulated and tested for performance. Comparisons are made with similar methods in order to verify the feasibility of the techniques developed. Towards the end of this study, the following systems are designed and proposed:

1. A computationally efficient robust watermarking system to enforce content protection and to prove ownership.
2. A low-complexity irreversible fragile watermarking system. Such a system would be effective in detecting modifications and tampering to the host video content.
3. A reversible fragile watermarking system which would be effective in authenticating a video content and/or its source. This method could also be useful in areas where any permanent change to the host video is considered unacceptable.
4. A computationally-efficient copy detection system based on the inherent characteristics of the video content. Such a system would be useful in tracking usage of a video content or for searching copies of a video within a large database.

The above systems can be considered to be the main focus of study within this work. In addition, development of these systems also involves analyzing the error resilience of H.264 video under a various attacks since illegal modification/editing of the video content by a digital pirate can be considered to be a problem of transmitting video content over an error-prone transmission network. Developing copy detection systems, in turn would involve studying those features within the H.264 encoder that are independent of various signal processing operations.

During the course of this study, the following original contributions were made. They are also included within the Bibliography.

Refereed Journal Publications:

1. M.A.Ali and E.A.Edirisinghe, “A semi-fragile watermarking technique for H.264/AVC using CAVLC,” *International Journal of Signal and Image Processing*, vol.1,No.3, Hypersciences, pp. 151-159, May 2010.
2. M.A.Ali and E.A.Edirisinghe, “Reversible watermarking using differential expansion on IPCM macroblocks in H.264/AVC,” *JNIT: Journal of Next Generation Information Technology*, vol. 2, no. 1, pp. 105-116, 2011.

Refereed Conference Proceedings

3. M.A.Ali and E.A.Edirisinghe, “Watermarking H.264/AVC by modifying DC coefficients,” in *International Conference on Cyberworlds, (CW09)*, (Bradford,UK), pp. 241-245, 7-11, Sept. 2009.
4. M.A.Ali and E.A.Edirisinghe, “Improved watermark payload capacity using DE on IPCM macroblocks in H.264/AVC,” in *5th International Conference on Computer Sciences and Convergence Information Technology (ICCIT10)*, (Seoul,South Korea), pp. 594-599, Nov.30-Dec.2, 2010.
5. M.A.Ali and E.A.Edirisinghe, “Multi-layer watermarking of H.264/AVC video using differential expansion on IPCM blocks,” in *IEEE International Conference on Consumer Electronics (ICCE11)*, (Las Vegas,Nevada,USA), pp. 53-54, Jan. 9-12, 2011.

Submitted Publications to Refereed Journals

6. M.A.Ali and E.A.Edirisinghe, “Multi-layer reversible watermarking for H.264/AVC video.” Submitted to *Signal Processing:Image Communication at Elseiver*, May 2011.
7. M.A.Ali and E.A.Edirisinghe, “Efficient spatiotemporal matching for video copy detection in H.264/AVC video.” Submitted to *IEEE Transactions on Multimedia*, May 2011.

1.5 Thesis Layout

For clarity in presentation, this thesis is arranged as follows: Chapter 2 explains the H.264 standard and some of its novel features. This chapter also briefly explains those features that were exploited within this work in order to design DRM systems. Chapter 3 introduces the concept of DRM in general and video content in particular. This chapter highlights the different aspects of DRM and the various methodologies that could be employed to enforce each of these aspects. Chapter 4 proposes a computationally-efficient DRM methodology for content protection and proof of ownership. This method is based on the robust watermarking technique. An irreversible fragile watermarking system for hard content authentication is introduced in Chapter 5. The performance of the designed algorithm under various attacks is also studied. Chapter 6 presents a fragile watermarking method that is reversible in nature. The need for DRM methods based on reversibility; and an application scenario is also included within this chapter. Chapter 7 takes a different approach to DRM by proposing a method based on content-based copy detection. This chapter also justifies the use of this method when compared to other methods such as watermarking. Conclusions are presented in Chapter 8 which highlights the contributions made within this work and also suggests a list of improvements/modifications that can be explored further. Bibliography is included at the end of the thesis.

Chapter 2

The H.264 Standard

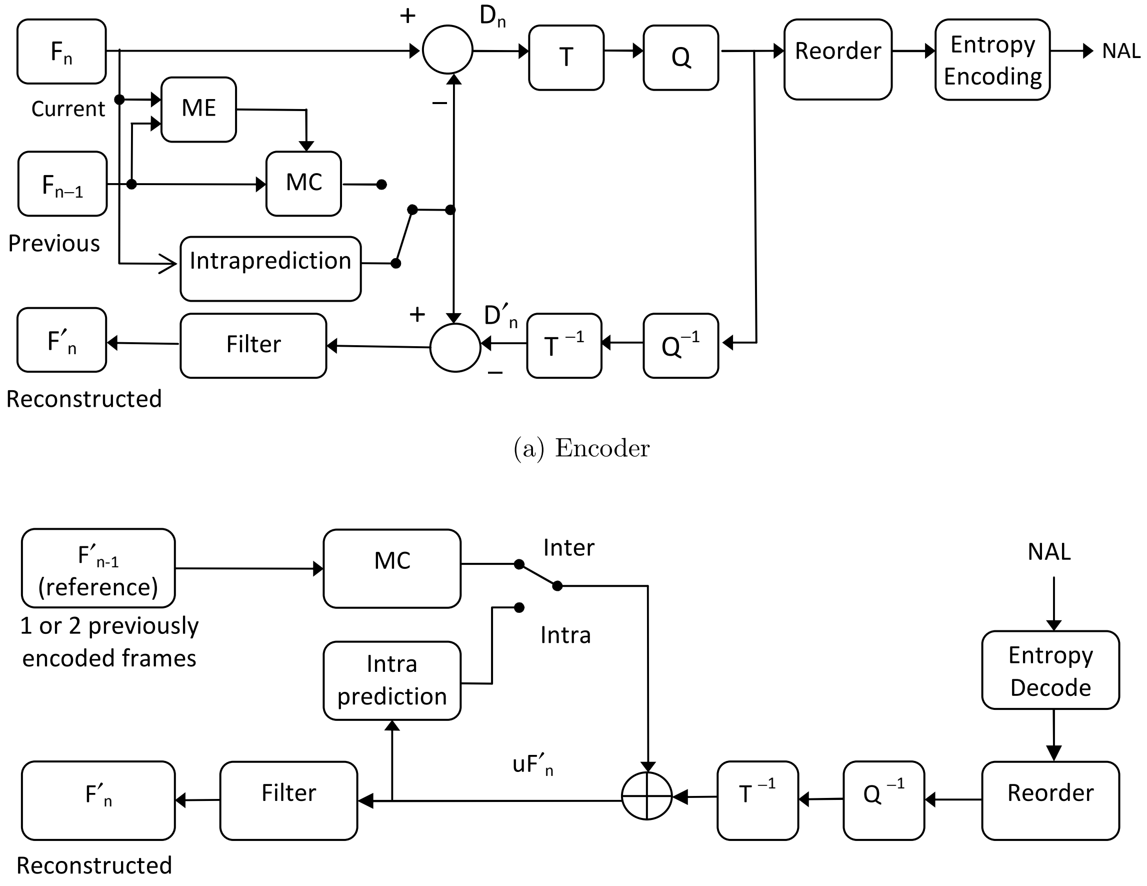
This chapter presents an overview of the H.264 video coding standard developed by the Joint Video Team. The overview discusses not only the functioning of the standard in general but also highlights those features that have been exploited within this work.

2.1 Overview

In early 1998, the Video Coding Experts Group (VCEG) ITU-T SG 16 Q.6 issued a call for proposals on a project named H.26L. The aim was to double the coding efficiency which effectively meant that for a given level of quality, the bitrate would be halved in comparison to the video standards available at that time. It was intended that the new standard would also cater to a large variety of applications. The first draft design was adopted in October 1999. By December 2001, VCEG and the Moving Picture Experts Group (MPEG) ISO/IEC JTC 1/SC 29/WG 11 formed a *Joint Video Team* (JVT), to finalize the draft of a new video coding standard. In March 2003, the standard was formally approved as H.264/Advanced Video Coding (AVC) [2, 3].Later,encouraged by the significant improvements in video compression capability, in January 2005, the JVT also standardized a Scalable Video Coding (SVC) [4] extension of the AVC. SVC is a very attractive solution to the problems posed by the characteristics of modern video transmission systems.

Since H.264 was designed for a large variety of applications ranging from conversational services on mobile networks to high quality video-on-demand, there was a need for flexibility and customizability. In contrast to earlier video coding standards, H.264 only defines the syntax of the encoded video bitstream and a method to decode the bitstream. There is no standard encoder or decoder but only compliant codecs. Nonetheless, most of the functional blocks of an H.264

compliant codec are also present in earlier standards with the exception of the deblocking filter. A simplified block diagram of the H.264 encoder and decoder is shown in Fig.2.1. As can be seen, the encoder has a forward path and a reconstruction path which is similar to that of the decoder.



(a) Encoder

(b) Decoder

Figure 2.1: Block diagram of the H.264 CODEC

The encoder supports a number of features to ensure enhanced coding efficiency and robustness to data errors/losses along with flexibility of operation over a variety of network environments. Some of the notable features to ensure the above characteristics are improved prediction methods, improved transform and entropy encoding methods and a new bitstream syntax structure.

As a result of the improved performance, H.264 finds acceptance in a broad spectrum of applications. For instance, the quality of television broadcast over satellite can be improved significantly. In mobile telecommunications, the cost of transmitting and receiving streaming video can be reduced due to the resulting lower bitrate.

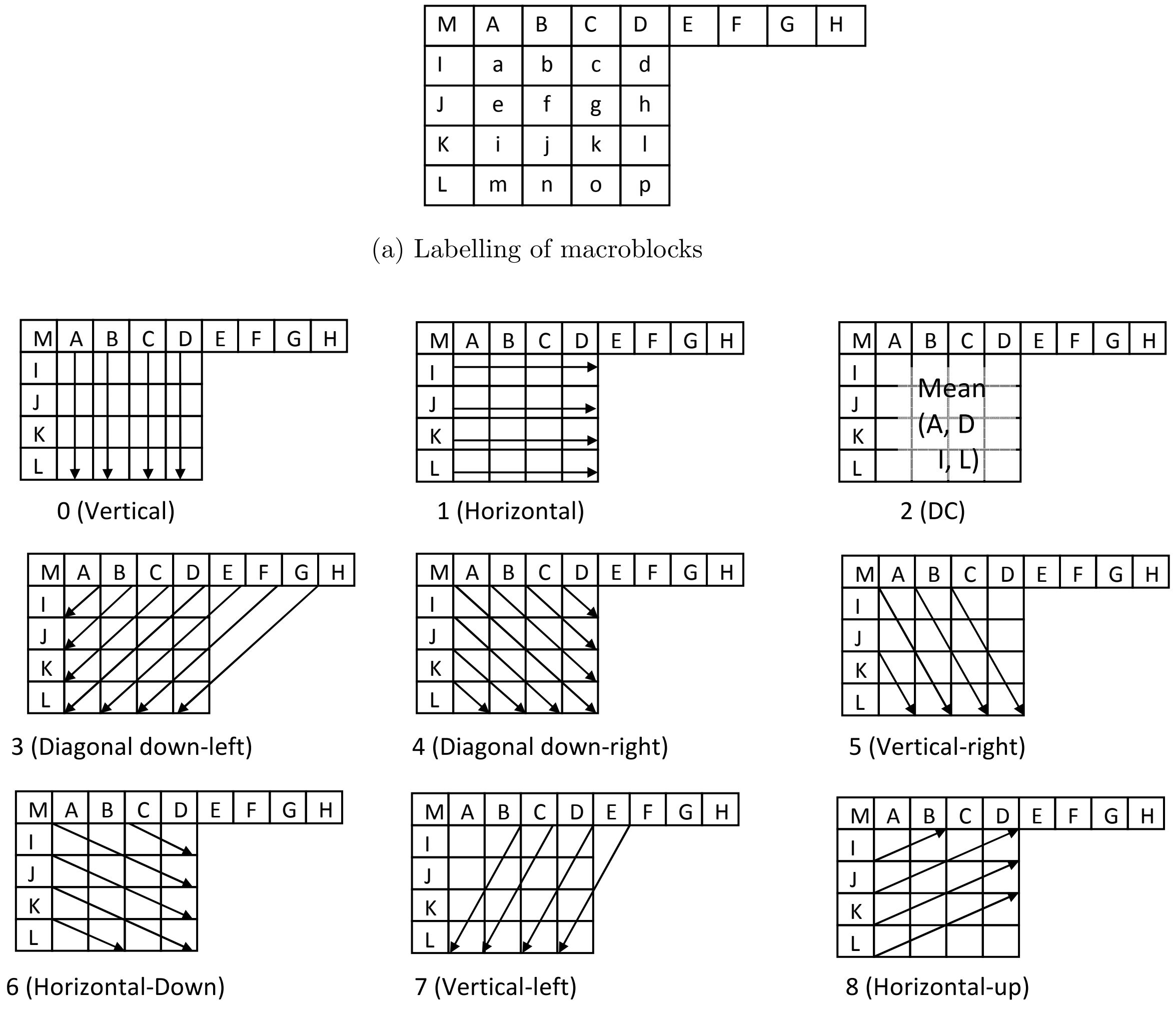
Thus the standard offered an all-around improved performance and compression ratio. In order to ensure that the algorithms developed within this work did not compromise on the performance of the above mentioned features, it was essential to understand their functioning. The following sections thus explain those features of the H.264 codec that were utilized in order to design the digital rights management algorithms.

2.2 Intraprediction

A coded picture consists of a number of *macroblocks*. These macroblocks are arranged in slices where a slice is a set of macroblocks in raster scan order. Macroblocks can be I, P or B. I macroblocks are predicted using decoded macroblock samples from within the current slice. P and B macroblocks on the other hand, are predicted from slices belonging to previously coded pictures named *reference picture(s)*. This section discusses the intraprediction methodology within H.264 while the next section discusses the features of interprediction.

Intraprediction [3] is one of the many new features incorporated in the H.264 standard in order to improve the compression efficiency. Unlike earlier standards which did not have any prediction within their I-frames, the H.264 standard supports intraprediction which means that sample values of macroblocks within I-frames are predicted from already transmitted neighboring macroblocks of the same frame. The predicted macroblock block is normally termed as prediction block P. The luminance values in P can be formed either using intra 4×4 or intra 16×16 block prediction mode. The intra 4×4 mode is used in the detailed and high motion areas of the frame while the intra 16×16 mode is used in the smooth and the stationary areas of the frame. There are a total of 9 prediction modes for intra 4×4 luminance block, 4 modes for 16×16 luminance block and 4 modes for the chrominance components. The different prediction modes and the direction of prediction for a 4×4 luminance block are shown in Fig.2.2. Labels *a-p* are the macroblocks that are to be predicted using previously encoded and reconstructed blocks labelled,*A-M* [5].

In certain situations, not all samples from *A-M* may be available (for instance, they might be a part of another slice). In such situations, only samples that are available within the current slice are used for prediction. This allows independent decoding of slices. Mode 2 (i.e. DC prediction) is modified depending upon which samples from *A-M* are available, however other modes are chosen depending upon the condition that the prediction blocks are available (within the same slice). The only exception is when blocks *E, F, G* and *H* are not available in which case the block values are copied from block *D*.



(a) Labelling of macroblocks

(b) Prediction modes for luminance

Figure 2.2: Intraprediction modes for 4×4 macroblocks [5]

The choice of the prediction mode is made on the basis of the sum of absolute errors (SAE). The H.264 encoder calculates the SAE for P under all the 9 modes and chooses the mode with the smallest SAE as the best prediction mode.

As mentioned above, there are four modes for 16×16 intraprediction. These are shown in Fig.2.3 below. Modes 0 to 2 are self explanatory while in mode 3, a linear 'plane' function is applied to the values obtained from the upper and left-hand pixel samples (H and V respectively). This mode is most appropriate for regions where the luminance varies smoothly.

Thus the H.264 standard supports a wide range of intra-prediction methods for the I-frame. They improve the coding efficiency while at the same time maintaining a good perceptual quality of the frame.

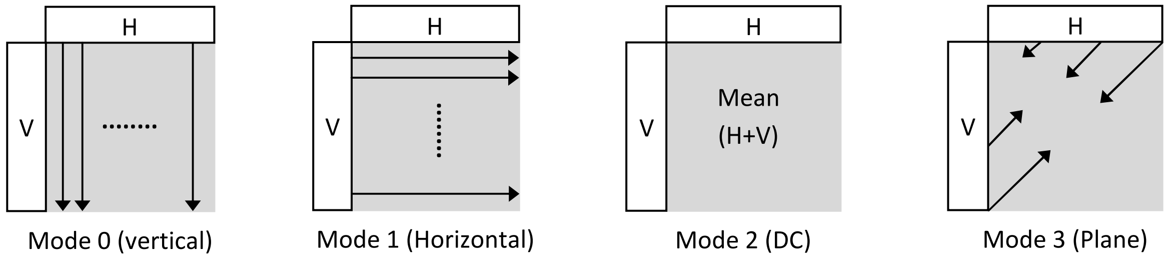
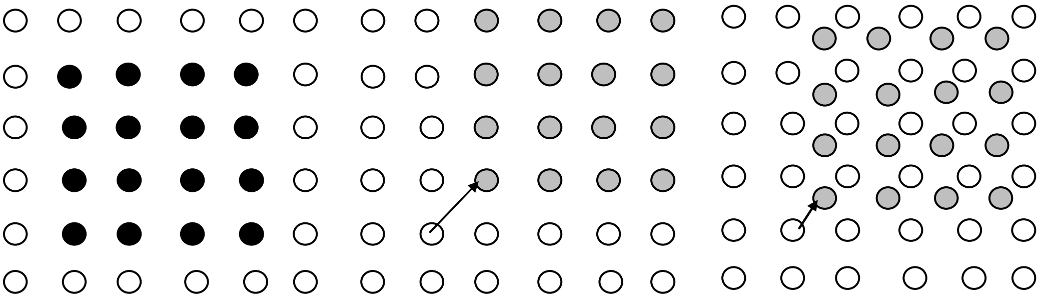


Figure 2.3: Intrapredication modes for 16×16 macroblocks [5]



(a) 4×4 in current frame

(b) Reference block: vector 0.2cm (1,-1)

(c) Reference block: vector (0.75,-0.75)

Figure 2.4: Interprediction modes for luminance components [5]

2.3 Interprediction

In addition to intraprediction, within P and B-frames, some macroblocks are predicted using motion compensation. For better prediction accuracy, these P macroblocks are further partitioned. Each macroblock partition in an inter-coded macroblock is predicted from an area of the same size in a previously encoded reference picture using block-based motion compensation. The offset between the two areas need not be exactly on the pixel resolution. H.264 takes into account such a possibility and offers quarter sample resolution for the luminance component and one-eighth sample resolution for the chrominance components. This is done by applying interpolation to nearby coded samples. Figure 2.4 illustrates this concept.

In Fig.2.4c, it can be seen that the actual pixel positions do not exist but in fact have been interpolated using nearby integral pixel positions. As mentioned earlier, for block-based motion compensation, a variety of block-sizes are supported within the H.264 standard. For a *P* macroblock, luminance block sizes of 16×16, 16×8, 8×16 and 8×8 are supported. When the 8×8 block size is chosen, an additional syntax element is transmitted which specifies whether the corresponding 8×8 block is further subdivided into partitions of 8×8, 8×4, 4×8 or 4×4. This improves the accuracy of the motion-compensation block within the H.264 encoder.

In addition to the above mentioned macroblock types, a P macroblock can also be coded either in the SKIP mode or the DIRECT mode [6]. These modes work under the assumption that there is a high spatiotemporal correlation between the motion vectors of adjacent macroblocks or frames. The SKIP mode corresponds to the spatial correlation while the DIRECT mode corresponds to the temporal correlation. Under these modes, the motion information for the current macroblock is derived from previously encoded information corresponding to adjacent macroblocks or frames. This eliminates the need to transmit either the motion vector or the quantized prediction error but only the index numbers of the macroblocks\frames referrred. *P_skip* macroblocks are very efficient and economical in depicting large areas with no change or slow constant motion such as panning.

B slices can be encoded and reconstructed similar to P slices, including the SKIP and the DIRECT modes. However, due to the bipredictive nature of B slices, motion vectors of a B macroblock could be pointing to two different references. These references are maintained as *List 0* and *List 1*. Having two reference lists further improves performance. The copy detection method proposed in Chapter 7 is based on recognizing 4×4 intrapredicted macroblocks within the I-frames and the *P_skip* and *B_skip* macroblocks within the P-and B-frames respectively.

2.4 Transform Coding

The resulting effect of spatiotemporal prediction, as shown in Fig.2.1 is the residual, D_n. This residual can be coded in 3 ways. The 4×4 array of luminance DC coefficients obtained from 16×16 prediction and the 2×2 array of chrominance DC coefficients are encoded using Hadamard transform. A DCT based transform is applied on 4×4 blocks of residual data after motion compensated prediction or intraprediction. The transform though based on the DCT has a few differences: It is an integer transform i.e. all operations are carried out using integer arithmetic without loss of decoding accuracy. Using integer arithmetic means that it is possible to ensure zero mismatches between the encoder and the decoder inverse transforms. The core part of the transform can be implemented using only additions and shifts. A scaling multiplication is integrated into the quantizer, reducing the total number of multiplications. The inverse quantisation (scaling) and inverse transform operations can be carried out using 16-bit integer arithmetic with only a single multiply per coefficient, without any loss of accuracy. Finally, applying the smaller 4×4 transform on macroblocks ensure better visual quality by reducing the ringing effects around areas having high detail/texture. Further details about the transform techniques implemented within the H.264 standard can be found in [3]. The robust watermarking technique proposed in Chapter 4 utilizes the DC coefficients generated as a result of applying DCT transform on the 4×4 blocks of residual data to embed the payload.

2.5 Entropy Encoding

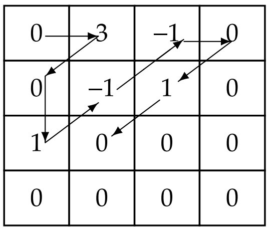
The H.264 standard supports two types of entropy encoding: Context-based Adaptive Binary Arithmetic Coding (CABAC) and Context-based Variable Length Coding (CAVLC) [7]. Since the technique proposed in Chapter 5 proposes a fragile watermark based on CAVLC, this section briefly outlines this technique. As the name suggests, entropy encoding within CAVLC proceeds by taking into account the context of the neighbouring blocks. It has a lower compression efficiency than CABAC and is lossy but also has a lower computational complexity. The CAVLC is used to encode the residual, zig-zag ordered 4×4 blocks of quantized coefficients. There are a number of characteristics of such a block that CAVLC exploits:

1. The high frequency coefficients at the end of the zigzag scan are mostly sequences of ±1. They are labelled as Trailing Ones(*T1s*).
2. Most of the coefficients of the block are zero after prediction, integer transform and quantization.
3. The number of coefficients in a block is correlated to the number of coefficients in the neighbouring left-hand and upper previously encoded blocks.
4. The coefficients are larger at the start of the zig-zag scan and smaller towards the end.

CAVLC takes advantage of these features by using run-level coding to compactly represent a long sequence of zeroes. Also, the choice of VLC look-up table to encode the values of the non-zero coefficients is adapted depending on recently-encoded values. CAVLC proceeds to encode the block of quantized coefficients (shown in Fig.2.5) as follows:

1. The coefficients are scanned in a zig-zag manner.
2. The total number of non-zero coefficients (*TotalCoeffs*) and the number of *T1s* are encoded together as *coeff-token* i.e. *coeff-token* =*TotalCoeffs*+ *T1s*. *TotalCoeffs* can be between 0 to 16. *T1s* can be between 0 to 3. If there are more than three *T1s* then they are assumed to be non-zero coefficients.
3. A look-up table is used to encode *coeff-token.* The choice of the look-up table depends on the number of non-zero coefficients in the left-hand (N_L) and upper (N_U) previously encoded blocks. The N_L and N_U values are used to evaluate $N = (N_L + N_U)/2$

4. Depending on the value of N, there are four choices regarding the look-up table as shown in Table 2.1. The first three are variable length code tables while the last one is a fixed length code table.
5. The sign of the *T1s* is encoded as *trailing_ones_sign_flag*.
6. The level (value) of the remaining non-zero coefficients is encoded as *level* in the reverse order. Again the choice of VLC tables to encode each level changes based on the magnitude of each successive coded level. There are seven tables to choose from as shown in Table 2.2 and the choice of the table is made in the following way:
   (a) Start by choosing *Level_VLC0*. However, if there are more than 10 coefficients and less than 3 *T1s* then initialize choice to *Level_VLC1*.
   (b) Encode the highest frequency non-zero coefficient using the chosen VLC table.
   (c) If the magnitude of this coefficient if higher than a pre-defined threshold, choose the next VLC table.
7. The total number of zeroes after the first non-zero coefficient is encoded as *total_zeroes*.
8. The number of zeroes preceding a non-zero coefficient during a reverse zig-zag scan is encoded as *run-before* in the reverse order with two exceptions:
   (a) There are no more zeroes left to encode.
   (b) *run-before* need not be encoded for the lowest frequency non-zero coefficient



0	3	−1	0
0	−1	1	0
1	0	0	0
0	0	0	0

Zig-zag scanned:0,3,0,1,-1,-1,0,1,0.............

TotalCoeffs = 5

T1s = 3 (actually 4)

Coeff_token = 8

Figure 2.5: An example of the CAVLC entropy encoding process

Table 2.1: Choice of table look-up for *coeff-token*

N	Table for *coeff-token*
0,1	Num-VLC0
2,3	Num-VLC1
4,5,6,7	Num-VLC2
8 or above	FLC

Table 2.2: Threshold to increment choice of table to encode level

Current VLC Table	Threshold to increment Table
Level-VLC0	0
Level-VLC1	3
Level-VLC2	6
Level-VLC3	12
Level-VLC4	24
Level-VLC5	48
Level-VLC6	N/A (highest table)

2.6 Bitstream Structure

As mentioned in Section 2.1, the H.264 standard only defines the syntax of the H.264 encoded bitstream and a method to decode it. This essentially implies that only the decoding process is "standardized" by imposing a restriction on the syntax and the structure of the resulting bitstream. Such a restriction would allow the encoding entity complete freedom and flexibility to configure the encoder to conform to specific applications. The encoder could be tuned to produce an encoded H.264 bitstream that has the right balance of quality, compression factor, implementation cost etc. in order to suit the application. This also ensures that every compliant decoder adhering to the standard would be able to decode an encoded bitstream, if it has been constructed according to the syntax.

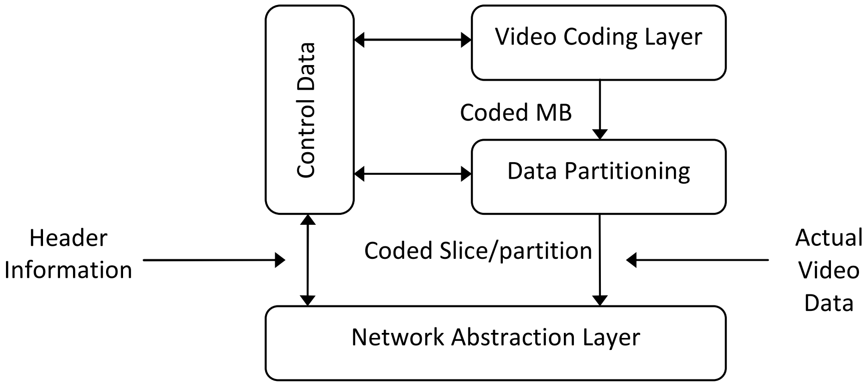


Figure 2.6: Bitstream generation within the H.264 encoder

Since the H.264 standard was meant to cater to a wide array of applications over different networks, it was imperative to design a syntactical structure that could handle this variety. To ensure adaptability and customizability, the actual video content and its header information were separated into two different entities. An H.264 encoded bitstream is basically composed of the Network Abstraction Layer (NAL) and the Video Coding Layer (VCL). As Fig. 2.6 shows [3], the VCL is the basic entity containing the actual video data which gets embedded within the NAL. The NAL formats the VCL information and attaches a header to it which contains control information pertaining to the video being encoded. In addition, the purpose of the NAL is to represent the VCL and the header information in a format that is suitable for transmission over a variety of transport layers and for storage.

The NAL unit is a logical data packet within an H.264 bitstream. Each packet contains an integral number of bytes. Within each NAL, the first byte is the header byte which describes the payload within the NAL unit. The remaining bytes are the payload as indicated by the header. The NAL structure may vary depending on whether the encoded video is being transmitted over a packet- or bitstream-oriented system. In a bitstream-oriented environment, it may be possible that only a partial NAL unit has been delivered. In that case, it may be necessary to identify NAL unit boundaries, not only for decoding but also for bitstream alignment. This is usually achieved by prefixing each NAL unit with a 3 byte code known as the *start code prefix.* They act as unique identifiers indicating the start of a new NAL unit. In a packet-oriented environment, the underlying protocol itself encapsulates the NAL units within a packet and attaches a unique identifier. Thus NAL units do not need to carry a start code prefix.

NAL units are further classified as VCL or non-VCL units. As these terms suggest, VCL NAL units contain data corresponding to video samples while the non-VCL NAL units do not contain any video data. Rather they contain associated additional information such as *parameter sets* that apply to a large number of VCL NAL units and other supplementary information. This optional information is not required to decode the video but can be used to enhance the usability of the decoded video signal.

An *access unit* is a set of NAL units combined together in a specified form. Decoding of each access unit gives one decoded picture. Thus each access unit is self-contained i.e. all the NAL units within it compose what is known as a *primary coded picture*. Within a primary coded picture, a set of VCL NAL units can comprise a *slice* or a *slice data partition* that represents an independent part of a video picture.

Finally, if the picture being coded is the last within a sequence then an *end of sequence* NAL unit may be appended, however if the picture is the last within the entire NAL unit stream then an *end of stream* NAL unit is appended to indicate the end of the stream.

It is clear that the H.264 bitstream is designed around self-contained NAL units. The concept of NAL units offers the facility to map the H.264 VCL data into a variety of network transport protocols and file formats. This promotes flexibility, simplicity, customizability and network friendliness of the resulting bitstream. A more detailed discussion on the above concepts and other features can be found in [8,9].

2.7 Conclusion

This chapter presented an outline of the H.264/AVC standard which was a joint collaboration between ITU-T VCEG and the ISO/IEC organizations. The standard supports many notable features that distinguish it from earlier standards including an enhanced and more accurate motion prediction, smaller block size integer-based transform and context-adaptive entropy encoding. The VLC based bitstream also makes the standard very flexible, network friendly and customizable. It has been claimed in [3] that the usage of these novel features can lead to a savings of up to 50% in the resulting bitrate for a comparable perceptual quality with reference to earlier standards.

However, the methodologies adopted to implement these features within the H.264 encoder make it very complex. From the point of the view of designing DRM systems for the standard, this becomes a drawback since any modification to the encoder in order to incorporate a DRM algorithm might make it more intricate and also compromise on its compression efficiency. Thus it is important to understand the functioning of these novel features so as to design DRM systems that do not have a significant detrimental impact on the performance of the standard.

Chapter 3

Digital Rights Management

This chapter discusses the concept of Digital Rights Management (DRM) when applied to digital content in general and then attempts to build a framework for video content in particular. Towards the end of the chapter, justification is provided for choosing some frameworks over others.

3.1 General Overview

Digital Rights Management (DRM) is an access control methodology that is employed by content creators/owners to protect and authenticate their content. DRM refers to the protection, distribution, modification, and enforcement of the rights associated with the use of digital content. The primary responsibilities of a DRM system include secure delivery of content, prevention of unauthorized access, enforcement of usage rules, and monitoring of the use of content [10, 11]. Usually, enforcing DRM would involve granting digital licenses rather than buying digital content. The license would dictate rules regarding the usage of the content such as frequency of access, view-by date, transfer constraints, restrictions on modifications and making copies. A basic DRM model would normally consist of four entities: the content provider, the distributer, the clearinghouse and the consumer. It is the content provider who usually decides the steps to enforce DRM. These steps could be taken either to protect the digital content, verify ownership or to authenticate content. Similarly, detection methods could be used by the content provider/distributer to verify copies of the original video content.

As explained in [10], a basic DRM system would have a set-up as shown in Fig. 3.1. Mostly, the content creator and distributor are a single entity hence broadly speaking; a DRM process model could have three entities. A content creator who creates the content and applies the DRM rule , a license creator who creates the license and attaches it to the content and finally the consumer who complies with

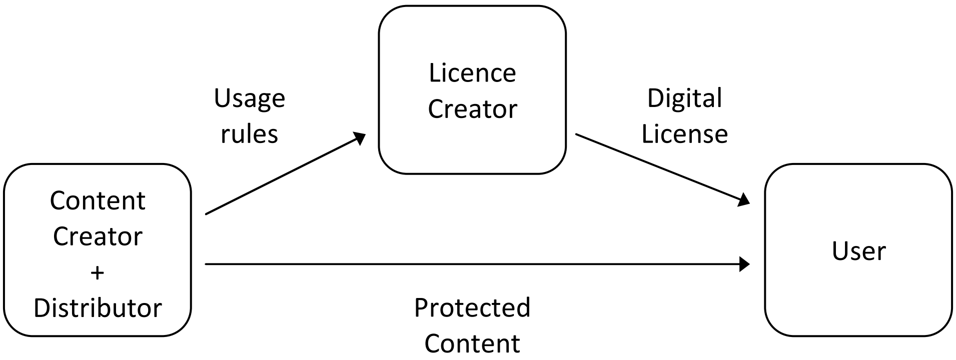


Figure 3.1: A simplified DRM process model

the license and goes on to consume the content.

A DRM process model works as follows: The content creator encodes and encrypts the digital content prior to distribution. The protected content is made available via a content distribution server. The license is created, attached to the content and made available by the license creator for consumption. A would-be consumer would download the content from the distribution server but would require a license to be able to decrypt the content and view it. The consumer would then request the license creator for a license. The license creator would identify and record the user and then charge him depending upon his usage request. After completion of the payment, the license creator would provide the consumer with a specific code that would decrypt the content. The user completes the transaction by decrypting and viewing the content. In certain cases, the license may be granted prior to or even simultaneously with the content. This approach could be more appropriate in situations where the consumer is encouraged to sample the content before the actual purchase.

DRM can be enforced using a number of mechanisms, as shown in Fig. 3.2, or in certain situations even a combination of these. For example, the video content could first be encrypted and then hashed using a cryptographic hash algorithm. At the receiving end, the hash would be used to match the digital signature of the encrypted video in order to verify its authenticity. After this verification, the video content is decrypted. Similarly, CBCD methods could be employed to detect and locate copies of an original video in a large video database or a longer video sequence. A robust watermark could then be used to claim ownership.

This chapter discusses the concept of content protection, proof of ownership, content/source authentication, and copy detection and how they are enforced using the methodologies shown in the figure below. Interestingly,cryptographic hash algorithms can be employed to enforce either of the two DRM mechanisms, as

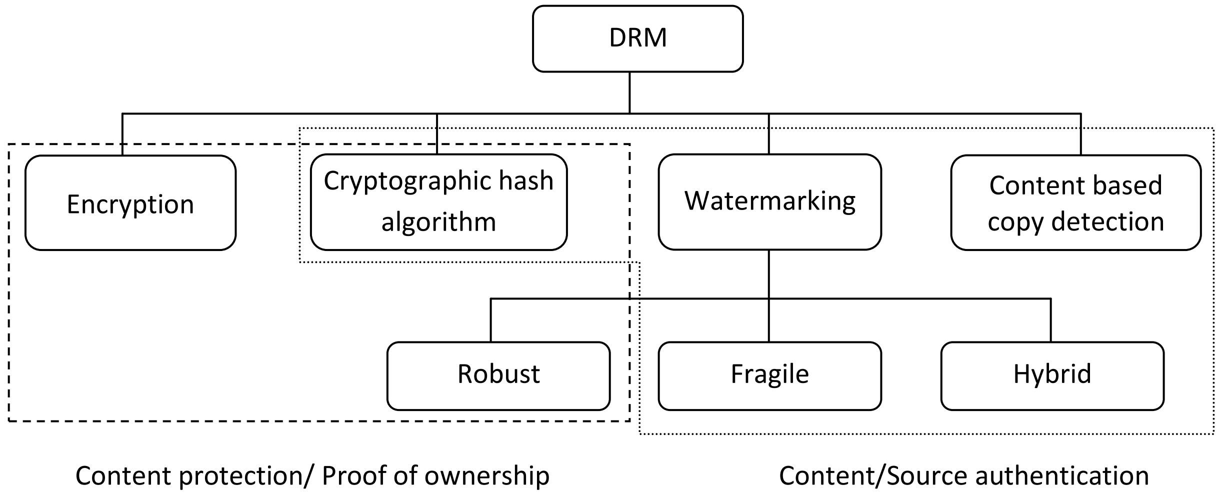

DRM
Encryption
Cryptographic hash algorithm
Watermarking
Content based copy detection
Robust
Fragile
Hybrid
Content protection/ Proof of ownership
Content/Source authentication

Figure 3.2: Techniques employed to enforce DRM

depicted in Fig.3.2.

Encryption and cryptographic hashing are related and but are often incorrectly used interchangeably. However, both of them produce an encrypted form of the original data that has to be decrypted before consumption and are therefore more appropriate for content protection. Watermarking and CBCD methods in contrast, retain the perceptual form of the original data and are more useful for content authentication. The following sections present a brief discussion of these methods when applied to video data.

3.2 Encryption

Video content encryption, termed as video scrambling prevents access to the content by distorting the video data such that it appears unintelligible to a viewer without prior descrambling. Descrambling can be done by compliant decoders which recognize the code before playback. Descrambling can also be performed if the viewer has the appropriate key which he/she can use to decrypt the video at the receiver end and view the content. However, this method involves extensive key management and distribution where unique keys have to be made available to each and every user. A basic encryption mechanism for video is shown in Fig.3.3

Some of the earliest methods to encrypt video can be found in [12, 13]. However, encryption as a content protection mechanism immediately runs into problems. Scrambling of video data is usually done in the spatial domain. This would drastically modify its statistical properties making it very difficult to compress. The obvious way out would be to compress the data prior to scrambling as proposed in [14–17] and depicted in Fig.3.3. Further, encrypting the data and then

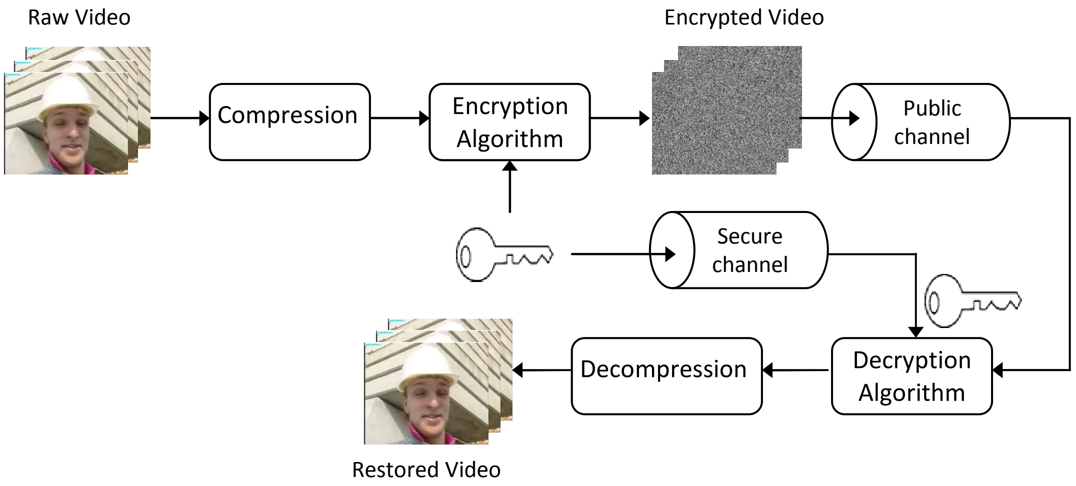


Figure 3.3: Techniques employed to enforce DRM

decrypting it before playback would lead to a significant processing overhead since cryptographic algorithms are usually complex computationally. In fact, decompression and real-time delivery of TV or cinema quality digital video in itself is a very challenging problem and constitutes its own area of research. Finally, the high computational complexity makes encryption algorithms unsuitable for low-power portable devices.

However, attempts have been made to design computationally efficient video encryption algorithms. In the context of H.264 video, there are a few methods worth mentioning. Zou *et al.* [18] proposed an encryption scheme which functions during the entropy encoding stage of the H.264 encoder. They partially encrypt the slice data and claim to preserve the network-friendliness and compression efficiency of the standard. Park and Shin [19] propose a selective encryption scheme wherein they only encrypt the intraprediction modes, the motion vector difference values and the sign bits of the texture data. They claim that their algorithm remains lightweight. There are a few other methods that encrypt H.264 video data in the compressed domain. Iqbal *et al.* [20] proposed a scheme wherein they selectively encrypt slice data partitions within the H.264 bitstream. Depending upon the level of encryption desired, higher or lower number of slices are chosen to be encrypted. They also claim that this selective encryption reduces the computational overhead.

It is clear that the advantages that can be offered by video scrambling methods are limited by the constraints of their own computational complexity. Particularly, when it comes to streaming video content over bandwidth-constrained networks and viewing them using low-power portable devices, encryption becomes a very significant hindrance.

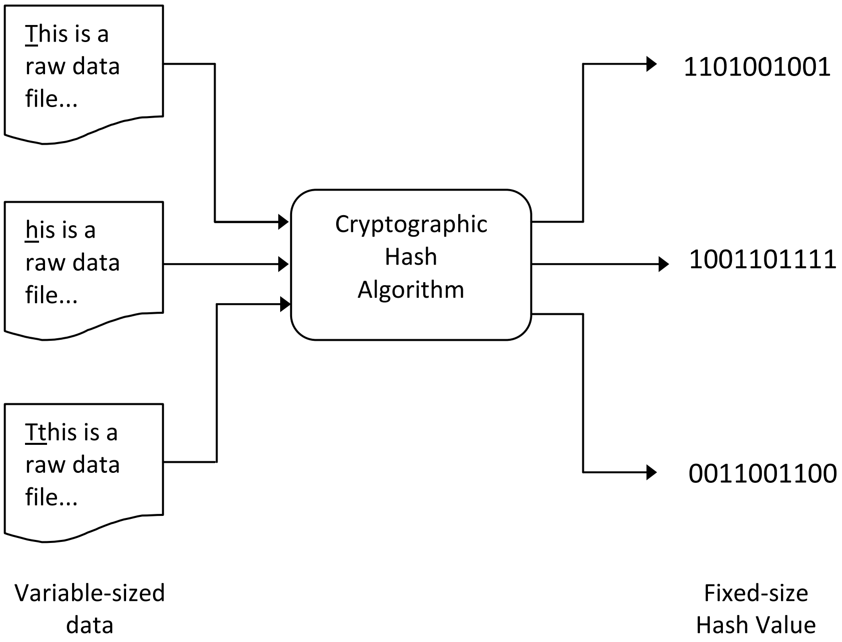

This is a raw data file...
his is a raw data file...
Tthis is a raw data file...
Cryptographic Hash Algorithm
1101001001
1001101111
0011001100
Variable-sized data
Fixed-size Hash Value

Figure 3.4: A simplified model of a cryptographic hashing system

3.3 Cryptographic Hash Algorithms

In cryptographic hashing, a hash function based on a cryptographic algorithm is applied to a variable-sized digital content in order to generate a smaller fixed-size hash value. This hash value is unique and serves as a fingerprint for the digital content. Further, these hash values are extremely sensitive. A change of even a single bit in the original content will change the resulting hash value. This makes them very effective in detecting changes to the original data. The working of a cryptographic hashing system is shown in Fig.3.4 where the underlined alphabet depicts the difference in each of the data files.

As can be seen, a change of even a single alphabet leads to different hash values. A typical cryptographic hashing system thus, should have the following properties:

1. For any given cryptographic hash function H and data d, it should be straightforward to calculate $h = H(d)$ where h is the hash value. However, for a given h, it should be very difficult to find d such that $h = H(d)$.
2. For any given data d, if the hash value is h; then modifying d to d' should also lead to h'.
3. For any two data sets d_1 and d_2, if $d_1 \neq d_2$; then always $H(d_1) \neq H(d_2)$

The first property is termed as *preimage resistance* while the second and third properties are termed as *collision resistance* [21]. However, the high sensitivity of hash values, as pointed out in the second and third properties turn into a drawback when it comes to data that has a lot of redundancy, such as multimedia data. For instance, a raw video and its compressed version would be very different in terms of their binary data but are similar perceptually. In such cases, obtaining different hash values for both them is clearly meaningless. Thus hashing methodologies that are capable of generating same hash values for perceptually similar video content but at the same time capable of detecting more drastic changes are required. This category of hashing methods is termed as robust or perceptual hashing systems [22]. Initial video hashing methods had their roots in hashing systems designed for images. In these methods, the hashing function was applied frame-by-frame. It is obvious that these methods are unable to exploit the temporal redundancies present in video and hence would be vulnerable against temporal resynchronization, frame rate change and frame dropping etc.

A few perceptual methods have however been proposed that deal specifically with hashing video data by taking into account the spatio-temporal characteristics. Oostveen *et al.* [23] designed a method to obtain a video hash by applying 2×2 spatiotemporal Haar filters on the randomized block means of the luminance component. Coskun *et al.* [22] proposed two video hashing methods based on DCT. One of them was based on the classical basis set while the other was based on the randomized basis set. Even though they claim robustness against signal processing attacks and transmission errors; they also admit that the former method lacks security since different video samples may give the same hash value and thus result in a collision. Collision usually happens if the hash function is not well designed, and would result in flagging false positives when it comes to detecting copies.

Recently a few hashing algorithms have been designed specifically for H.264 video. These hashing algorithms would be robust to underlying changes made to the content by the H.264 encoder. Ramaswamy and Rao [24] proposed a hard video authentication and sender verification algorithm based on a cryptographic hash. They extract the DC and the first two quantized AC frequency coefficients from every macroblock within every frame of a GOP. These are then hashed to generate a fingerprint for that GOP. They claim that this would produce a unique fingerprint for every GOP and thus would capture the spatio-temporal characteristics of the video sequence. Wang *et al.* [25] proposed a scheme wherein the encryption keys are generated based on a cryptographic hash function. They encrypted the intraprediction mode, the motion vector difference and the quantized coefficients. A hash function was then applied to the encrypted data to produce a hash value. They claimed that the proposed scheme is efficient computationally and the encryption/hashing process hardly affects the resulting video quality.

3.4 Drawbacks of Encryption and Cryptographic Hash Algorithms

From the above discussion, the underlying fact to emerge is that encryption and cryptographic algorithms will almost always lead to a processing overhead and time delay when it comes to transmission and playback of video data. A DRM system based on encryption/cryptographic hash algorithms can suffer from one or all of the following drawbacks:

1. Require a strong mathematical foundation in order to design an effective system.
2. The security of the techniques depend solely on the key that encrypts the data. It is usually said, "*Lose the key and you effectively lose the data*", since there is no alternative to recovering the data without decrypting it using the key that was used to encrypt it.
3. Cryptographic hash algorithms are very sensitive. Even a change of a single bit is enough to change the hash value. This makes them ineffective for multimedia data wherein the raw data may be drastically different (due to compression or other operations) but is perceptually similar.
4. They are computationally very demanding and will almost always consume a significant amount of computing resources when in execution.
5. Encryption is not effective in protecting content after it has been decrypted. An authorized user, after decrypting the video can easily modify, duplicate and re-distribute the video. A similar problem can also be foreseen in the context of authentication.
6. Integration of a cryptographic algorithm to an existing system is usually quite difficult. This means that implementing a cryptographic algorithm on an existing system mostly leads to unwanted side effects such as affecting operational performance.

It is clear that the encryption and cryptographic hash algorithms are inefficient when it comes to enforcing DRM for video in general and H.264 video in particular. This is due to the fact that the H.264 standard is widely used for streaming video applications on devices running on limited power and computing resources. This constraint, in itself, is enough to make these DRM approaches unfeasible.

Watermarking, in turn, provides a better alternative for content protection and authentication in the context of H.264. Since a watermark is embedded within the video content and effectively becomes a part of it, it is permanent. Even after the video content has been authenticated and an authorized owner identified, the watermark is still a part of the content. In addition, a watermark can be embedded in such a way that it can not only provide authentication and protection but also indicate attacks and tampering on the video content. Watermarking techniques are also generally computationally efficient and do not place a burden on the video codecs. Similarly, CBCD methods exploit characteristics that are inherently unique to video content in order to detect copies. No external methods/techniques are required in order to match and locate copies. Again, this fact makes CBCD much more feasible for H.264 video as compared to encryption and hashing.

Thus, watermarking and CBCD methods are more viable alternatives to implementing DRM, specially for H.264 video. These two methods are discussed in the following sections.

3.5 Watermarking

Watermarking is a class of data embedding technique wherein the data to be embedded has a close relationship with the host i.e. the content it is being embedded into. In the context of images and videos, the relationship could be in the form of pixel values in the spatial domain, the transform method used, quantization parameter value or the entropy encoding method used to compress the host. However, embedding the watermark inevitably leads to distortions and introduction of artifacts. Usually, higher the watermark payload, higher will be the amount of distortion introduced. Thus it is highly desirable that distortion/artifacts introduced as a result of the embedding should be at least, visually imperceptible. Consequently, human visual model systems could be employed to reduce the effect.

However the underlying fact is that since the watermark is embedded within the digital content and effectively becomes a part of it; it is permanent. Even after the content has been authenticated and an authorized owner identified, the watermark is still remains an integral part of the content. A watermark can be embedded in such a way that it can not only provide authentication and protection but also indicate attacks and tampering on the video content. Watermarking techniques are also less complex computationally.

Watermarking as a technique to prove ownership was proposed for the first time more than 60 years ago [26]. Since then watermarking algorithms have been employed to protect all kinds of data, multimedia or otherwise. Mathematically, any watermarking algorithm for images/video can be defined as:

$$I_W = I_O + W$$

where I_O is the original information vector that can be pixel values, transformed coefficients or any other information about the content. I_W is the watermarked information vector while W is the watermark to be embedded. As can be seen from

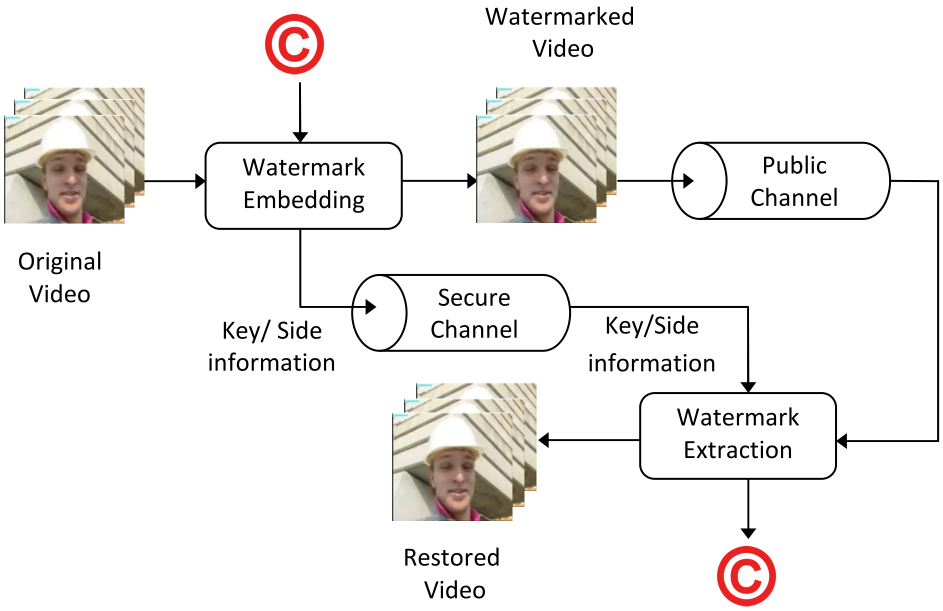


Figure 3.5: A simplified watermarking model

the above equation, the watermark signal is considered to be the difference between the original content and the watermarked content, no matter how it actually gets embedded. A simplified model of a typical video watermarking system is shown in Fig.3.5.

As can be seen from Fig.3.5, in contrast to encryption (see Fig.3.3), the watermarked video retains its perceptual form. In fact, watermarking digital content involves taking into consideration several factors. It involves tradeoffs between the amount of modification made to the data on one hand and the degree of immunity to host signal attacks and visual quality degradation on the other. A large amount of signal modification can often lead to significant degradation in the host signal which is obviously not desirable. Thus it is essential that the modification be made to the host data up to a certain level so that the watermark is extracted/detected within the desired error probabilities. This is largely referred to the rate-distortion characteristics (R-D) in literature. Usually, optimum R-D characteristics are obtained by deciding the domain in which the watermark algorithm is to function. Domains are classified as: spatial domain/pixel domain, transform domain/frequency domain or the compressed/bitstream domain.

Spatial domain watermarking techniques directly modify the pixel values/ sample points of the digital content. Some of the earliest algorithms based on this technique are presented in [27–30]. Transform domain watermarking techniques embed the watermark after DCT, DFT or wavelet operations on digital data. Some frequency coefficients in the transform domain are selected to carry the watermark. The selection is made under a watermarking rule. Some very popular frequency domain watermarking techniques have been reported in [31–33]. Compressed domain watermarking algorithms are designed to perform embedding after the quantization and entropy encoding stage. At this stage the quantized frequency coefficients are encoded as variable length codes (VLCs) and fed into the bitstream. The watermarking algorithm thus modifies the bitstream to embed the watermark. Compressed domain techniques are relatively newer than spatial and transform domain techniques. Some landmark compressed domain techniques are presented in [34, 35].

Depending on the way the watermark is inserted and depending on the nature of the watermarking algorithm, the watermark detection method can take on two approaches. In the first approach, the watermark has to be extracted in its exact form. This is referred to as 'watermark extraction'. In the second scheme, it may only be necessary (or possible) to detect whether a specific watermark signal is present. This is referred to as 'watermark detection'. Both of these approaches have their own application domains [36]. The first approach finds use in authentication systems. In such a system, a robust cryptographic hash function of the host content is computed and the resulting value is embedded as the watermark. To authenticate the watermark, the embedded hash value is extracted and compared against the computed hash value. A difference in hash values would mean the content has been modified. The second approach finds usage in areas such as broadcast monitoring stations. In this scenario, a monitoring station might be set up by a broadcast company, the purpose of which would be to check if the local stations broadcast their content without paying the relevant charges. This can be done by only detecting the presence of a watermark. There is no need to actually extract the watermark.

In situations where the watermark is to be extracted rather than detected, there are again two possibilities. The first possibility is that the original content is also sent across to the receiver side, possibly via a secure channel, and is compared with the watermarked and possibly corrupted content. The difference between the two contents is effectively the watermark. Such a technique is referred to as informed watermarking. But this technique is not always practical. First of all, the content owner may not want to distribute the unwatermarked content just to check the existence of the watermark. Secondly, it makes the overall system more complicated since the watermarked content may require some pre- processing before a comparison can be made. For instance, translation, rotation angles and scale factors may need to be adjusted; error correction may be required and so on. The second possibility is to have a technique that does not need the original content to extract the watermark. Such a technique is called a 'blind' watermarking technique and works by using a secret key to identify the watermark. Usually, the key is the same as that used to embed the watermark. Blind watermarking techniques are more practical, have a lower complexity and hence preferred over informed watermarking techniques. A few informed watermarking techniques are reported in [37, 38] while some of the earlier blind watermarking techniques can be studied in [39, 40].

Watermarking techniques are also classified as robust or fragile. Robust watermarking is more suited for content protection and copyright ownership while fragile watermarking is more appropriate for content authentication. There is also a third category of semi-fragile watermarks. As the name suggests, semi-fragile watermarks are somewhere in between robust and fragile. Such watermarks are sensitive to most of the attacks but are resistant to common processing tasks such as compression.

It should now be apparent that robust watermarks are the preferred means of protecting video content. To authenticate video content however, fragile watermarks are more appropriate. They are designed in such a way that they get altered or distorted even under the most common signal processing operations. A perturbed watermark is an indication that the host data has been altered, damaged or modified by an unauthorized user. Video content containing a damaged watermark will thus fail the authentication process. Fragile watermarks have found wide-spread acceptance in areas such as protecting images archived in a database [41, 42] and applications where the aim is to ensure that the image or the video has not been fabricated to falsify events, such as in news agencies [43]. A few other applications include medical images and forensics, legal evidence and espionage [44].

Fragile watermarks usually compete with cryptographic hash-based signature systems. However, fragile watermarks offer two distinct advantages over hash algorithm-based authentication systems. First, the watermark becomes a part of the host data in contrast to being an additional data set. Secondly, hash signature-based authentication systems can detect alterations but cannot identify the location of the alterations. In contrast, fragile watermarks can not only detect that the host data has been altered but also highlight the tampered locations.

However, the fact cannot be overlooked that watermarking essentially involves modification of the host data. This fact can sometimes be unacceptable or plain impractical. In such situations, content-based copy detection methods offer a better alternative.

3.6 Content-Based Copy Detection

Content-based copy detection (CBCD) is another method to protect digital content. Generally, multimedia data contains enough unique information that can serve as its fingerprint in order to identify it. The fingerprint can be used to detect copies, either within a large database or in case of video, within a longer video clip as well. A basic CBCD system is shown in Fig. 3.6.

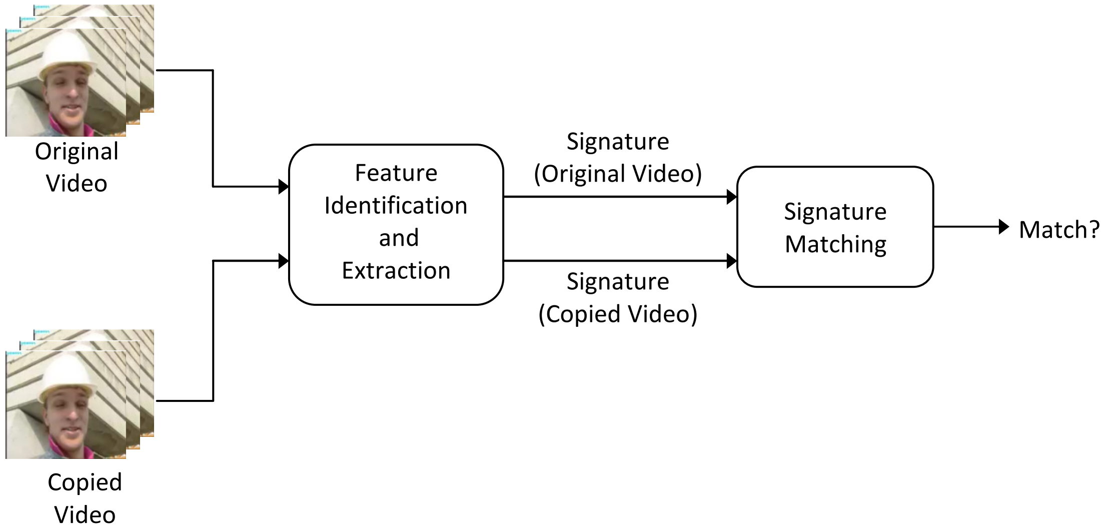


Figure 3.6: A basic content-based copy detection system

As can be seen, no additional information is required, in contrast to encryption and watermarking, in order to enforce DRM. The most popular statement in support of CBCD systems is that "the media itself is the watermark". Another added advantage is that the signature need not be extracted before the media is actually distributed as was the case in watermarking. Finally, since no additional processing is required, CBCD methods are computationally efficient which makes them suitable for applications where computing resources are limited or at a premium. Usually CBCD methods are complimentary to watermarking. For instance, after the CBCD based method detects illegal copies, the rightful owner of the content can use a watermark in order to prove ownership.

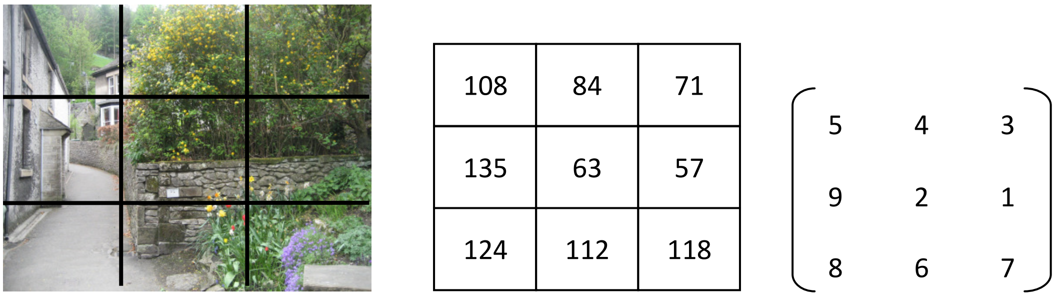
CBCD methods were initially proposed for images with the most straightforward method being correlation-based, which calculated the sum of pixel differences [45]. As the term suggests, this method calculates the difference between two images pixel-wise. More formally, given an image A and its copy B with pixel intensity values $\{I_A^1, I_A^2, ..., I_A^n\}$ and $\{I_B^1, I_B^2, ..., I_B^n\}$ respectively, the copy detection mechanism would calculate a correlation-based distance parameter D as:

$$D = \frac{\sum_{i=1}^{n} \mid I_A^i - I_B^i \mid}{n}$$

D is then compared to a specified threshold value. A value of D lesser than the threshold would indicate a copy. However, it is clear that matching each pixel intensity value will be computationally intensive. The obvious way to make it more efficient would be to match average intensities of each corresponding macroblock rather than each pixel. However, simple matching of pixel intensities is not a robust copy detection mechanism simply due to the fact that even a single outlier pixel or block value can render the detection mechanism ineffective. Further, non-linear intensity variation within pixel or block values also makes the technique unsuitable.

Consequently many other methods were proposed in order to overcome these drawbacks. They were based on wavelet-transforms [46], colour histogram intersection [47] etc. However, one of the most effective methods to emerge in recent years is based on ordinal measures. This method, originally proposed in [48], works by dividing an image into $n \times n$ equal-sized regions and then calculating the average intensity within each region. The division makes the method independent of input image size. The average intensities obtained from each region are arranged ordinally in a one-dimensional rank matrix which acts as the signature for the image. This rank matrix was found to be independent of most signal processing operations and was thus effective in detecting copies of images that had undergone operations like changes in luminance, contrast, sharpening, median filtering, image resizing, letter- and pillar-box etc. Figure 3.7 and 3.8 depict an example of this methodology.

As can be seen, irrespective of the change in luminance of the image, the method generates the same ordinal matrix and hence the signature as the original image. This makes the technique very effective in detecting modified copies. This method was adopted by many other copy detection approaches such as the one proposed in [45] wherein the ordinal ranking matrix was generated for DCT coefficients and found to be more effective than simple ordering of intensity values.



(a) Original image divided into 3×3 blocks

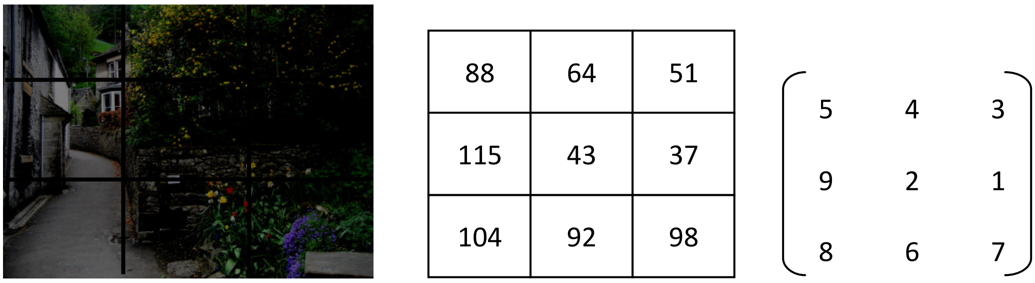
108	84	71
135	63	57
124	112	118

(b) Average intensity value of each block

$$\begin{pmatrix} 5 & 4 & 3 \\ 9 & 2 & 1 \\ 8 & 6 & 7 \end{pmatrix}$$

(c) Ordinal matrix

Figure 3.7: Ordinal signature generation for an original image



(a) Copied image divided into 3×3 blocks

88	64	51
115	43	37
104	92	98

(b) Average intensity value of each block

$$\begin{pmatrix} 5 & 4 & 3 \\ 9 & 2 & 1 \\ 8 & 6 & 7 \end{pmatrix}$$

(c) Ordinal matrix

Figure 3.8: Ordinal signature generation for a copied image with reduced luminance

The effectiveness of this method in detecting copies of images implies that it can be modified to detect copies of video sequences as well. In fact, a number of approaches have been developed as can be found in [49–54]. The copy detection method proposed in Chapter 7 is also based on this approach.

3.7 Conclusion

Techniques to protect content from unauthorized copying and proving ownership have been around for close to 70 years now. However it is only recently that they have assumed a much more significant role. Enforcing DRM could be achieved using a number of techniques that include encryption, cryptographic hash algorithms, watermarking and CBCD. However, this chapter showed that when it comes to protecting video data, the first two, though quite effective, suffer from a number of drawbacks. This makes them impractical to be implemented for a number of applications, for instance when accessing streaming video over portable and handheld devices. In such application scenarios, watermarking and CBCD methods offer a better alternative. Watermarking, as a technique, has been proven to be effective in not only protecting video content and proving ownership but also in detecting unauthorized modification and tampering. CBCD-based methods have been found to be very effective in detecting copies of video, not only within a large database but also in detecting copies of short video clips in longer video sequences. This approach is very useful in tracking the usage of video content. CBCD differ from other methods in that they not require any supplementary information to detect copies.

As pointed out in Section 3.4, besides the distinct advantages offered by watermarking and CBCD methods, there is one more reason due to which it was decided to focus this work on developing DRM techniques based on them. Since the H.264 standard was chosen as the platform on which the algorithms would be tested, it was important to keep in mind its application domain. This standard also supports streaming video over cellular networks which mean that it is designed to work in a bandwidth constrained environment and on devices having limited computing resources. Thus it was important that the DRM techniques developed are not only effective but also computationally efficient. Consequently, the following chapters propose DRM techniques for H.264 video based only on watermarking and CBCD.

Chapter 4

Robust Watermarking

This Chapter presents a method for content protection and to prove ownership of digital media. These aspects are realized by employing a novel method of embedding a robust watermark within an H.264/AVC coded video. The proposed method works in the transform domain by embedding the watermark in the residual DC values of the 4×4 intra-prediction blocks. The DC values that are to contain the watermark bits are chosen randomly. The set of random values that are used to select the DC values are considered to be the 'key'. This key is made available at the receiver side in order to extract the watermark and verify ownership. Experimental results show that the technique is transparent with minimal effect on the host video sequence. Further, since only the DC coefficients are utilized to contain the watermark, the embedded watermark is resistant to compression and other similar modifications.

4.1 Introduction

Robust watermarking systems work towards preventing unauthorized users from destroying the secret code that has been embedded within the multimedia information in order to uniquely identify it. Robust watermarking systems usually come under two categories of attacks, termed as *fair* or *unfair* attacks. Attacks that make use of publicly available information, such as the algorithm used to embed the watermark, are termed as fair attacks while those that attempt to extract secret information such as the key used to embed the watermark, are termed as unfair attacks. Robust watermarking systems are expected to withstand fair attacks and degrade graciously under unfair attacks. Thus, it is imperative that the amount of information to be kept secret has to be minimal. Usually, the only information to be kept secret is the watermark embedding and extraction methodology i.e. the key. The algorithm and its domain of functioning are usually made public.

More recently, maintaining transparency while embedding a watermark is assuming as much significance as robustness. This is due to the fact that even the most robust watermarking systems have a high probability of being compromised under certain powerful attacks such as collusion. Collusion attacks are a more serious problem when it comes to video and audio signals. There are two types of collusion attacks. If the same watermark is embedded in different data, the watermark data can be estimated from each occurrence and the average of those estimates will be a refined estimate. If different watermarks are embedded in the same data, several users can collude by averaging their decoded signals to reduce the strength of the watermark and possibly render it unreadable. If the same watermark is embedded in all the frames, the first type of collusion can be used to remove the watermark from different scenes. If a different watermark is embedded in each frame, the second type of collusion can be used to remove the watermark from correlated scenes.

Transparency ensures that the hacker/cracker does not suspect that the digital content is watermarked and hence doesnt think of launching attacks to remove the watermark. One of the easiest methods to maintain high transparency is by performing watermark embedding in the same spectral components as that of the host data. This usually involves the mid and the low frequencies. But the drawback of most transform based techniques is that they involve the host in the detection process, which may not be feasible in all situations and applications. In addition, frequency domain watermark embedding techniques are not very robust. Even relatively weak attacks such as filtering and compression can lead to a considerable loss of the watermark data. Loss of watermark information due to unintentional attacks such as compression is highly undesirable since compression is one of the most essential video/image processing step and is widely applied on almost all digital media. This Chapter thus presents a robust transform domain watermarking technique that is not only transparent but also robust against compression.

4.2 Literature Survey

Since many standards such as JPEG, MPEG, H.263 and H.264 make use of the DCT, many watermarking algorithms have been proposed that work in the DCT domain. One of the first algorithms was proposed by Koch *et al.* [55]. They used a pair of mid frequency coefficients to embed a watermark bit. Bors and Pitas [56] proposed modifying the mid range DCT coefficients using linear DCT constraint or a circular DCT detection region. The selection of the frequency components was done using a Gaussian network classifier. Frequency masking of DCT blocks was proposed as a watermarking technique by Smith and Comisky [56] as well as by Swanson *et al.* [57].These techniques of watermark embedding in the DCT coefficients has been extended to H.264 AVC video standard as well.

Video watermarking is done taking into account the three factors of payload, quality and robustness. These are conflicting requirements and a tradeoff is made, usually by concentrating on one of them more than the other two. Noorkami and Mersereau [58] proposed a method wherein the watermark is embedded in the quantized AC residuals of the luma component of 4×4 intra-predicted macroblocks. Their technique embedded one watermark bit per quantized AC residual. The security of the algorithm is based on the randomness of the selected blocks. Wu *et al.* [59] proposed a blind watermarking technique which used a pair of predicted DCT coefficients within 4×4 blocks to embed 1-bit of a watermark. The embedding locations of DCT coefficients are switched from lower to higher subbands in a predefined order. Meerwald and Uhl [60] developed a robust watermarking framework for H.264/SVC wherein they add pseudo-random watermark bits to the residual blocks generated by the encoder. The modified residual block, after entropy encoding, is added to the bitstream. Gong and Lu [61] proposed a watermarking scheme that modified the DC coefficients of the luminance components within the residual blocks. They also applied a texture-based perceptual model in order to improve the perceptual quality of the resulting video while at the same time maintaining robustness. They claim that their algorithm can adaptively choose the watermark strength based on the characteristics of each residual block.

Most of the DCT based video watermarking techniques are limited to manipulating only the AC coefficients or the residuals. At the most, some of the low frequency AC coefficients/residuals are modified to contain a watermark bit. This work instead aims to use the DC coefficients of 4×4 residual blocks to contain the watermark.

4.3 Proposed Robust Watermarking Methodology

The proposed scheme embeds the watermark in the 4×4 DC residuals generated by the H.264 encoder. The DC values are read out from every residual macroblock as shown in Fig.4.1. The watermark is assumed to be a random sequence of binary values. The choice of DC residuals and the embedding within them is made as follows: An index pair [i, j] is randomly generated for every set of 4×4 DC residuals, where 1≤i, j≤4. $DC_{i,j}$ is then the chosen residual. The watermark bit

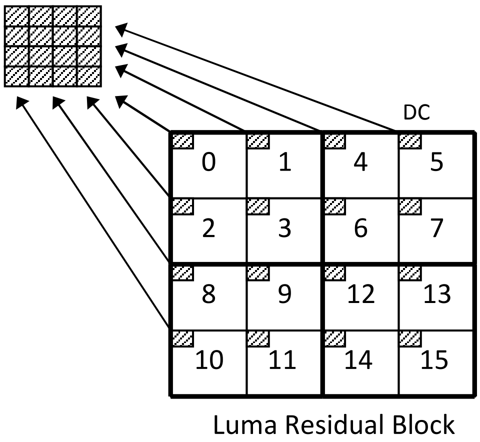


Figure 4.1: Extraction of residual DC coefficients from a 4×4 luminance macroblock

simply overwrites the Least Significant Bit (LSB) of $DC_{i,j}$. The set of randomly generated $[i, j]$ is the key.

As is obvious, on an average, half of $DC_{i,j}$ values will remain unchanged since the LSB being overwritten might be the same as the watermark bit being embedded. But even those DC values that have changed, the change is not significant. Only for those rare conditions where the chosen $DC_{i,j}$ is very small, will some change be noticeable. Further, the embedding of the watermark is not just limited to I-frames, as is normally done by earlier video watermarking algorithms, but involves the P and the B-frames as well. This leads to a higher payload capacity. Figure 4.2 indicates the stage within the H.264 encoder where the proposed algorithm intervenes to embed the watermark.

In Fig.4.2, D_n is the residual value of a 4×4 macroblock to be encoded. Before it is sent for transformation and quantization, the watermark embedding algorithm intervenes and embeds the payload within D_n. These modified residual blocks are then again handed over to the encoder wherein they are transformed, quantized and entropy encoded and transmitted. At the decoder side, these modified residual macroblocks are identified using the key (which could be sent across as supplementary information or via a secure channel) and the LSBs of the matched $DC_{i,j}$ are read out. The LSBs are reconstituted to build the extracted watermark. The extracted watermark can be compared to the original to find out the number of correctly extracted watermark bits. Figure 4.2 also illustrates that the proposed algorithm is spatial domain technique.

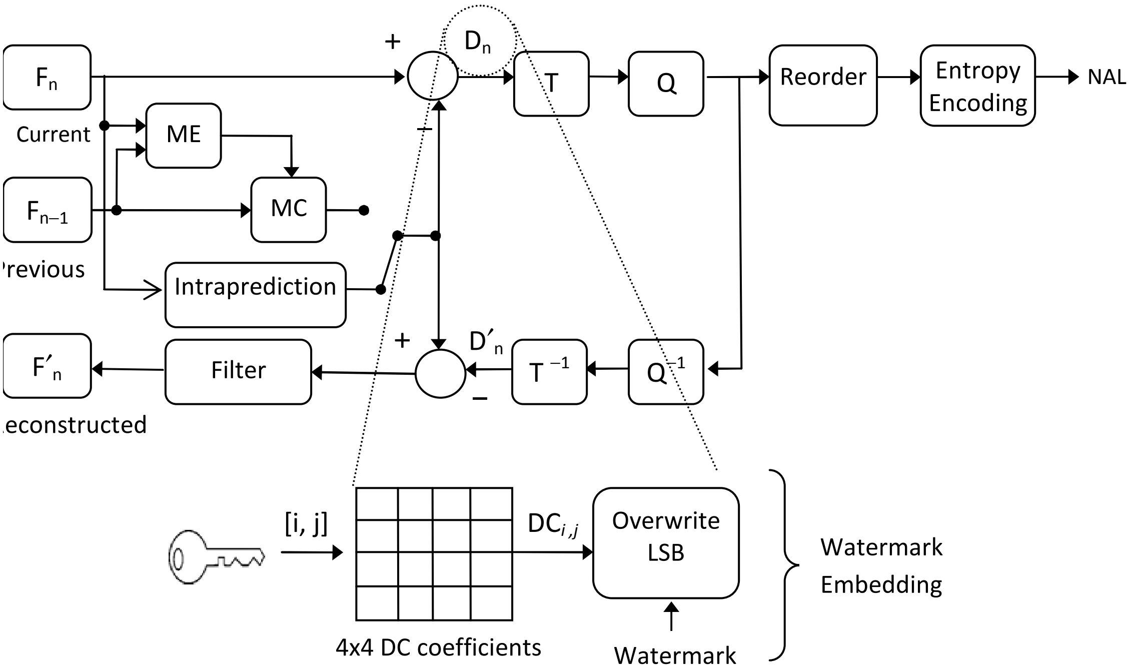


Figure 4.2: Watermark embedding within DC coefficients

4.4 Simulation Results and Performance Analysis

The proposed watermarking algorithm was implemented in the H.264 reference software version JM15.0 [62]. The watermark to be embedded was assumed to be a two-tone image consisting of a pseudorandom sequence of 1s and 0s $\{w(i), i{=}1, 2, ..., n\}$, where n is the watermark length and each $w(i)$ is either 0 or 1. Standard QCIF, 176×144 video sequences were used as the host. The frame rate was set at 30 frames per second. The software was run on a number of sample videos such as *Foreman*, *Coastguard*, *Miss America* and *Suzie*. Each of these samples was chosen as the host for watermark embedding, once with the rate control OFF and then with the rate control ON. The performance of the watermarking algorithm was checked, first with the number of frames equal to 3 and then increased to 16 to include at least one GOP.The results are presented in Tables 4.1*a*, 4.1*b*, 4.2*a* and 4.2*b* respectively.

With a 176×144 video, and 16×16 prediction, there is a total of 11×9=99 macroblocks per frame. Since 4×4 luminance coefficients are used to obtain the DC residuals, this will lead to a total of 16×99=1584 DC residuals per frame. Since one DC residual is chosen out of every 4×4=16 residual values to contain a watermark, each of the (1×99) DC residual values contain one watermark bit per frame. These calculations imply that a 3-frame video sequence would contain 1584×3=4752

Table 4.1*a*: PSNR and bitrate values for 3 frames with rate control OFF

Video	Without watermark			With watermark		
	PSNR(Y) (dB)	Total bits	Bitrate (kbps)	PSNR(Y) (dB)	Total bits	Bitrate (kbps)
Foreman	37.08	34624	346.24	36.93	34624	346.24
Miss America	41.13	12968	129.68	40.91	12968	129.68
Coastguard	35.68	37224	372.24	35.11	37224	372.24
Suzie	37.91	19864	198.64	37.41	19864	198.64

Table 4.1*b*: PSNR and bitrate values for 3 frames with rate control ON

Video	Without watermark			With watermark		
	PSNR(Y) (dB)	Total bits	Bitrate (kbps)	PSNR(Y) (dB)	Total bits	Bitrate (kbps)
Foreman	32.38	164.00	164.00	32.06	16400	164.00
Miss America	37.04	6576	65.76	36.79	6576	65.76
Coastguard	30.87	13184	131.84	30.25	13184	131.84
Suzie	33.97	8888	88.88	33.41	8888	88.88

Table 4.2*a*: PSNR and bitrate values for 16 frames with rate control OFF

Video	Without watermark			With watermark		
	PSNR(Y) (dB)	Total bits	Bitrate (kbps)	PSNR(Y) (dB)	Total bits	Bitrate (kbps)
Foreman	36.82	84976	159.33	36.54	84976	159.33
Miss America	40.80	22368	41.94	39.56	22368	41.94
Coastguard	34.87	111480	209.30	34.29	111480	209.03
Suzie	37.50	35392	66.36	36.90	35392	66.36

Table 4.2*b*: PSNR and bitrate values for 16 frames with rate control ON

Video	Without watermark			With watermark		
	PSNR(Y) (dB)	Total bits	Bitrate (kbps)	PSNR(Y) (dB)	Total bits	Bitrate (kbps)
Foreman	30.91	27560	51.67	30.22	27560	51.67
Miss America	39.43	25560	47.92	38.98	25560	47.92
Coastguard	34.87	111480	209.03	34.36	111480	209.03
Suzie	35.20	25760	48.30	34.63	25760	48.30

DC residual values.Out of these, only 99×3=297 DC residual values contain the watermark information i.e. only 6.25% of the DC residuals are modified to contain the watermark. Even though these payload values seem to be low, in a typical video sequence however, there would be thousands of frames and in such a case, the amount of payload that can be embedded will be very high.

The results in the above tables indicate that there is a very small change in the PSNR values, both with the rate control turned OFF and ON. The video sequences have been chosen in such a way that two of them are high motion sequences i.e. *Foreman* and *Coastguard* while the other two are low motion sequences. For the high motion sequences, the prediction will be less accurate and hence the residual values will be relatively high. This means that embedding a watermark into these residuals will lead to lesser degree of change in their values. Correspondingly, the PSNR values after embedding the watermark remain much closer to the unwatermarked values. This fact can be verified from the above tables. In contrast, the DC residuals will be quite small for low motion video sequences since the prediction will be more accurate and hence watermark embedding can lead to a greater degree of change. This results in PSNR values that are farther away from the PSNR values of the unwatermarked video sequence. This fact can also be verified from the above tables by examining the difference in the PSNR values of *Miss America* and *Suzie*. Thus, the proposed technique offers better performance for high motion video as compared to low motion ones.

When the rate control is turned ON, the quantization parameters change dynamically to adjust to the required bitrate. This means that the DC residual values may get changed as well to meet the bitrate requirements. However, even this constraint leads to imperceptible changes in the quality of the watermarked video. This proves that the proposed watermarking technique is highly transparent.

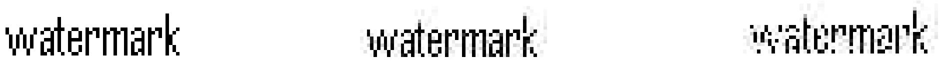
Finally, this algorithm is more robust than the algorithm proposed by Noorkami and Mersereau [58]. In their work, they utilized only the quantized AC residuals of the luminance component of 4×4 intra-predicted blocks to contain the watermark. When a very low bitrate is required, the quantization parameter will be set to be a very high value. This may lead to even some of the mid- and low-frequency AC residuals being dropped off and which may, in turn, lead to a loss of some watermark bits. The technique proposed in this paper is resistant to such a scenario since the DC coefficients are rarely modified as a result of re-compression. This is verified from Table 4.3 which shows that the proposed technique not only offers a higher payload per I-frame but is also more robust to re-encoding than the method proposed in [58]. This table also proves that the proposed technique has no effect on the bitrate of the encoded video in contrast to the method proposed

Table 4.3: Performance characteristics of the proposed technique

Video	Watermark bits		Re-compression recovery rate (%)		Bitrate increase(%)	
	Proposed method	Method in [58]	Proposed method	Method in [58]	Proposed method	Method in [58]
Carphone	99	44	73	58	0	0.80
Claire	99	22	88	83	0	0.44
Mobile	99	85	91	85	0	0.23
Mother	99	42	66	68	0	0.69
Table	99	38	71	62	0	0.31
Tempete	99	81	89	83	0	0.44

in [58].

To extract the watermark, the key was made available at the decoder. It was found that irrespective of the rate control being ON or OFF, almost the entire watermark could be successfully extracted. This verifies the theory that DC coefficients are largely unaffected by the changes in quantization parameters which in turn ensures the robustness of the watermark. Some of the extracted watermarks are depicted in Fig.4.3. Figure 4.3b shows the extracted watermark when the rate control (RC) is OFF and the quantization parameter is changed from 28 to 32. Figure 4.3c shows the extracted watermark with the RC turned ON. It can be seen that there is slightly more degradation in the quality of the extracted watermark with the RC turned ON than with OFF. This is due to the fact that with the RC turned ON, the encoder dynamically changes the value of the quantisation parameters to satisfy the upper limit imposed by the available bandwidth.



(a) Original Watermark (b) Watermark extracted with RC=OFF (c) Watermark extracted with RC=ON

Figure 4.3: Extracted watermark

4.5 Conclusion

Robust watermarking systems play an important role when the aim is to prove ownership of any digital media. This is due to the fact that they are able to resist any modifications that might be made to their carrier. This chapter presented a robust watermarking system that made use of the DC coefficients within 4×4 residual macroblocks generated by the H.264 encoder. The proposed technique was tested on a wide variety of both, high motion and low motion video sequences. Simulation results show that the embedded watermark is capable of surviving varying bitrate constraints and hence compression. Further, there is an imperceptible change in the quality of the host data and no increase in the resulting bitrate since the algorithm only replaces the LSB of a DC coefficient. This means that the technique is also particularly suited for low bitrate applications. The drawback of the proposed watermarking system is that it is ineffective against those signal processing applications that change the prediction mode since that would in turn lead to a change in the residual values and loss of the watermark. The results of this work appear in [63].

Chapter 5

Irreversible Fragile Watermarking

The advantage of digital content is that it can be easily copied, modified or edited. However, in many instances this could turn into a drawback whereby unauthorized users can illegally modify, copy and distribute the content. In such cases, it might be necessary to authenticate the content in order to detect whether it has been tampered with. One of the methodologies developed to authenticate digital content is fragile watermarking. The method proposed in this chapter is an irreversible watermarking technique while the method presented in Chapter 6 is a reversible watermarking approach. Both techniques were implemented within the JM Reference Software and experiments were conducted to test their performance.

5.1 Introduction

A fragile watermarking system has characteristics which are similar to any other watermarking system. However, the side information which is used to detect and extract the watermark plays a much more important role in a fragile watermarking system. The side information can be the "key" used to identify the embedding locations within the video, the watermark itself, or any other auxiliary information.The watermark detection system uses a statistical testing model to detect whether the image has been tampered with. If the detection model detects tampering then it is also desirable for it to detect the location of the tampering. There are a few other features that are desirable within a fragile watermarking system.They are briefly outlined in the following paragraph.

One of the most desirable, in fact, essential feature is perceptual transparency.This feature implies that the watermarked host data should remain perceptually similar to its original (un-watermarked) counterpart. Next, the watermark detection system should be unambiguous i.e. only the relevant key should be able to detect the watermark. Any other side information should fail to detect the existence of a watermark. The detection of the watermark should be blind i.e. the detection system should not need the original host data to successfully detect the watermark. The security of the key should be independent of the detection mechanism i.e. knowledge of the detection methodology should not reveal the key. A good fragile watermarking system should keep the key private while making the detection mechanism public. If both are linked then unauthorized users can use the detection mechanism to reveal the key, use it to remove the original watermark and embed their own. A more detailed discussion on the properties of a desirable fragile watermarking system can be found in [41,64,65]. However, the significance of one feature over the other depends on the application domain of the fragile watermark.

There are some attacks that are specifically targeted towards fragile watermarking systems. It is imperative to have an understanding of these in order to design improved systems. Most of these attacks, though fairly common, are usually quite potent in rendering the watermarking system ineffective. They include attacks that a fragile watermarking system may fail to detect. For instance, a new watermark may be carefully embedded so that it doesnt perturb the earlier watermark. The detector then falsely accepts the new mark as authentic. In some cases, an attacker may use brute force to try to completely remove the watermark. The owner of the content then has nothing to authenticate. Some attacks are launched on the authentication model itself rather than on the content in order to force it to accept a possibly tampered set of data as genuine. Further details of attacks on authentication systems can be found in [66].

The above discussion regarding the features of a fragile watermarking system and the attacks on them are equally valid for both, irreversible and reversible systems and hence also applicable to the method proposed in Chapter 6

5.2 Irreversible Fragile Watermarks

Fragile watermarking techniques have largely been ignored by the research community but it doesnt make their significance any lesser [67]. There exists a large class of applications that can benefit immensely by making use of fragile watermarking systems. Fragile watermarks can play a very important role in detecting unauthorized tampering and modifications within video as well. However, there exist only a few fragile watermarking systems that have been designed specially to cater to video. A quick look at the state-of-the-art shows relatively fewer fragile watermarking algorithms designed for the H.264 standard. There are two reasons, the first one being that the standard generates a highly compressed video stream. This leaves a very limited set of redundancies that can be exploited to embed a watermark. The second reason is that for H.264 video, content and copyright protection are considered to be of higher significance than authentication. Since the first reason is a hurdle which must be crossed even by content protection (robust watermarking) systems, it is actually the second reason which contributes towards a comparatively smaller body of literature that exists dealing with authenticating H.264 based content.

Some of the most popular fragile watermarking techniques for H.264 are highlighted as follows. Chen *et al.* [68] proposed a semi-fragile H.264 based video authentication method in which they used the block sub-band index and coefficient modulation of the quantized AC coefficients of the I-frame to embed the watermark. They claim that the video quality goes down only marginally and that they are able to locate the tampered location in the watermarked video frames. Kuo *et al.* [69] suggested a method wherein they exploited the fragility of the motion vectors to embed the watermark. They claimed that their technique maintained the perceptual quality of the host video. Wang and Hsu [70] proposed embedding a blind watermark in the residual macroblocks for H.264 video stream authentication. They used the last non-zero quantized coefficient of the residual block to contain the watermark information. The technique proposed in [69] promises the best rate-distortion characteristics among three other motion-vector based watermarking techniques. The algorithm proposed in [70] showed that the watermark can detect any tampering done due to recompression or GOP removal. After these two attacks the watermark extractor simply returned random noise.

Pröfrock *et al.* [71] presented a fragile, blind and erasable watermark that used some of the skipped macroblocks within the H.264 video. After selecting a set of skipped macroblocks, two popular watermarking techniques i.e. Least Significant Bit (LSB) and Quantization Index Modulation (QIM) were used to embed the watermark. The number of skipped macroblocks, their distribution and the number of watermark bits per block is decided by a special process. This technique works by performing partial entropy decoding to obtain the video data out of the bitstream, choosing the skipped macroblocks, embedding the watermark along with the encrypted hash value and a public key into the skipped macroblocks. These modified macroblocks are again re-inserted into the H.264 bitstream. Even though such a technique ensures that the watermark is truly blind, it leads to additional computational overhead since the partially decoded bitstream is to be re-encoded before transmission.

Most of the above mentioned techniques operate in the transform domain by modifying the coefficients before they are entropy encoded. There is clearly, a lack of techniques that operate during or after the entropy encoding stage. Fragile watermarking techniques perform best when they are implemented in the compressed domain. The reason being that the content is already highly compressed with almost no redundancy. Hence any kind of attack is enough to perturb the bitstream and the embedded mark thereby indicating tampering. Compressed domain watermarking techniques offer the advantage of being computationally efficient, not needing the original content for detection and if done right, being perceptually as well as analytically transparent. At the receiver side, the watermark can be extracted before decompressing the video.

The technique proposed in the next section works by embedding the watermark during the CAVLC entropy encoding stage thereby becoming a compressed domain approach and hence satisfying the above mentioned requirements.

5.3 Proposed Algorithm

The important aspect to consider within CAVLC entropy encoding is the number of non-zero coefficients (*TotalCoeffs*) and the number of *T1*s [see section 2.5]. This means that effectively there should no change in the value of *coeff-token*. Any change in the value of *coeff-token* would change the choice of look-up table for its neighbouring blocks and subsequently its VLC. This in turn, might affect the quality and bitrate of the resulting encoded video. Thus, to satisfy the criteria of transparency, any watermarking technique based on CAVLC should ensure that the value of *coeff-token* and hence the choice of look-up table remains unchanged.

The technique proposed here works by only modifying the last non-zero coefficient of the 4×4 block during the entropy encoding stage. The justification for choosing 4×4 blocks is that H.264/ AVC performs 4x4 transform in the complex and textured areas with detailed information. The human eye is less sensitive to any changes/modifications in such an area and hence embedding a watermark would ensure better transparency. In contrast, the encoder performs 16×16 transform in the smooth areas and those areas with less activity. Any changes in such areas would be easily noticeable and hence is not a suitable choice for watermark embedding.

Let the last non-zero coefficient be referred to as *last_coeff* from this point onwards. From section 2.5, it is clear that *last_coeff* can either be a *T1* or any other non-zero value, say, *c*. Since the proposed technique only modifies *last_coeff*, the modification could be either on a *T1* or on *c*. Further, since it is the highest frequency component, any modification to it would lead to imperceptible changes. At the same time, this coefficient is the most sensitive and any intentional or unintentional modifications of the block would perturb the watermark information thereby indicating tampering. The mechanism for embedding and extraction is given in Algorithm 1 and the location within the encoder where the technique has been incorporated is depicted in Fig.5.1

```
input  : $n$,the number of 4×4 blocks within an I-frame;
         $block_i$, the $i^{th}$ $4 \times 4$ block within an I-frame;
         $\{w(k)$,$k$=1,2.....$m\}$ the watermark sequence, where $m$ is the
          watermark length, each $w(k)$ is either 0 or 1 and $m$<<$n$;
         $S$, the set of $m$ random numbers between 0 and $n$;
output:  Random 4×4 macroblocks containing watermark bits
Watermark embedding:
for each element i within S do
    if $block_i$ = empty then                                  // all residuals==0
        skip ($block_i$);
        else readlastcoeff ($block_i$);
    if lastcoeff $== T1$ then
        if $w_k$== 0 then set $T1 = c$ else set $T1 = -1$;
    if lastcoeff $== c$ then
        if $w_k$== 0 and $c\%2 = 0$ then set $c = c$;
        else if $w_k$== 0 and $c\%2 = \pm 1$ then
            set $c = c + 1$;
        if $w_k$== 1 and $c\%2 = 0$ then
            set $c = c + 1$;
        else if $w_k$== 1 and $c\%2 = \pm 1$ then
            set $c = c$;
Watermark extraction:
for each element i within S do
    if $block_i$ = empty then                                  // all residuals==0
        skip ($block_i$);
        else readlastcoeff ($block_i$);
    if lastcoeff $== T1$==1 then
        $w_k$ ==0;
        else $w_k$==1;
    else if lastcoeff $== c$ then
        if $c\%2 = 0$ then
            $w_k == 0$;
        else if $c\%2 = \pm 1$ then
            $w_k == 1$;
```

Algorithm 1: Watermark embedding and extraction methodology

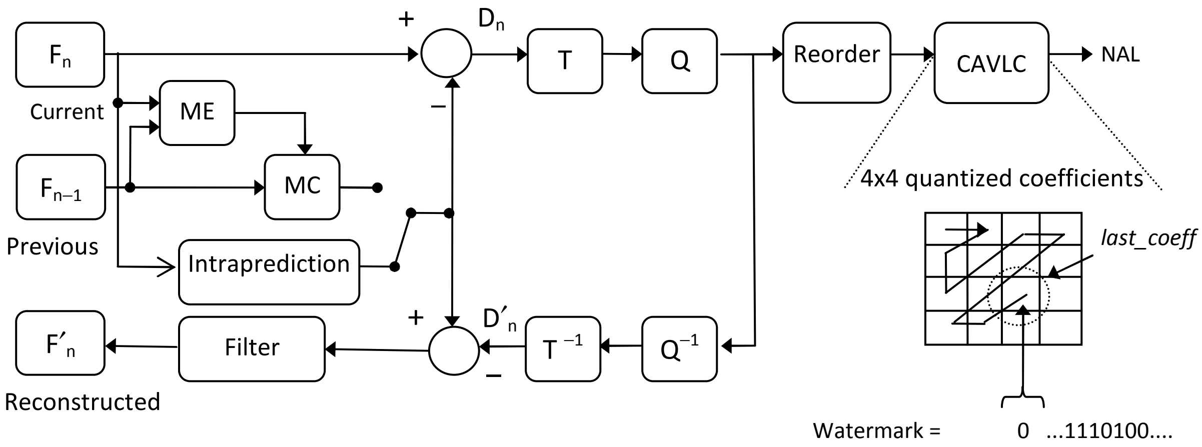


Figure 5.1: Domain of the proposed fragile watermarking approach

As can be seen from the above algorithm, theoretically, there is a 50% chance that *last_coeff* would remain unchanged. In addition, it can also be deduced that if there is no change in any of the VLCs then there would be no change in the bitrate. Even when c is modified, the technique ensures that the change is never more than 1 bit per c, thus the increase in the total bitrate is never too significant. This should ensure a high level of transparency. In addition, since the proposed algorithm involves simple mathematical operations, the watermark embedding and extraction module is not expected to add any significant computational overhead to the H.264 codec.

The choice of using the I-frames only to embed the watermark offers two very obvious but useful advantages. First, the significance of I-frames in a video sequence discourages a potential attacker from modifying since it will affect all subsequent frames predicted from it leading to errors such as drift thereby degrading overall video quality. Secondly, watermarking the I-frames offers the luxury of embedding a higher payload since very few 4×4 blocks are empty. The proposed technique ensures that all non-empty 4×4 blocks contain one bit of watermark information within c. In a 176×144 I-frame and assuming only 4×4 integer transform, there would be 1584 blocks per frame. The set of random values within S would specify the order in which the watermark bits will be embedded within these blocks. Theoretically, if every block were to contain a watermark bit then there would be 1584 watermark bits per frame which can be considered to be a significant payload.

The security of the algorithm can be increased by randomly choosing a subset of non-empty 4×4 blocks to embed the watermark bits rather than every 4×4 block. But this will be at the expense of a lower watermark payload. Hence a trade-off is to be made between the security of the algorithm and the payload size.

5.4 Simulation Results and Performance Analysis

The proposed watermarking algorithm was implemented in the H.264 reference software version JM15.1 [62]. For the sake of simplicity, the watermark to be embedded was assumed to be a two-tone image of size 26×26 pixels at 1 bit per pixel [see Fig.5.3a]. Standard QCIF, 176×144 video sequences were used as the host to embed the watermark. The frame rate was set at 30 frames per second, the GOP length was fixed at 16 and the rate-distortion optimization was switched off. The H.264 software was run on a number of sample videos such as $Foreman$, $Coastguard$, $Hall$ and $Container$. The performance of the proposed technique was evaluated under different encoding conditions such as using a wide range of quantization parameters. The computational complexity of the watermarking technique was evaluated along with the PSNR value comparison. The results at QP=30 are shown in Table 5.1. As can be seen, there is a negligible change in all the four parameters. Thus the technique satisfies the criteria of transparency. In fact, it can be observed that the increase in the number of bits is, at the most 16, particularly in the case of $Foreman$,*Container* and $Carphone$.

Another point of interest is the total encoding time. The increase in the total encoding time is barely 2 seconds more than the total encoding time of the encoder running without the embedding module. This proves that the proposed watermarking algorithm is computationally quite efficient since it uses simple operations like *mod* and *addition* for watermark embedding. The bitrate value also stays almost the same even after watermark embedding for all video samples.

The watermark is extracted during the CAVLC entropy decoding stage. The detector at the decoder side reads *last_coeff* as per the extraction technique explained in Algorithm 1 and extracts the watermark. It can be seen that the watermark is extracted before the frame is fully decoded and the original frame is also not required thus making this technique a blind one. The extracted watermark is compared with the original to check tampering. This is done by calculating the *Normalized Coefficient* (NC) as:

$$\rho = \frac{\sum\limits_{i=0}^{n}[(x_i - \bar{x})(y_i - \bar{y})]}{\sum\limits_{i=0}^{n}(x_i - \bar{x})^2 \sum\limits_{i=0}^{n}(y_i - \bar{y})^2}$$

where x_i and y_i are the pixel values of the original and the extracted watermark respectively and $\bar{x}$ and $\bar{y}$ are their mean values. It can be seen that $\rho = 1$ for a exact match between the embedded and the extracted watermark and 0 for a total mismatch.

Table 5.1: Performance characteristics at $QP = 30$

	Without watermark			
Video sequence	PSNR(Y) (dB)	Total bits (bits)	Total encoding time (secs)	Bitrate (kbps)
Foreman	35.75	643872	1082.66	120.73
Coastguard	34.84	1143992	1131.90	214.50
Hall	37.19	249432	1033.93	46.77
Container	35.98	175880	1058.14	32.98
Carphone	37.70	407360	1056.72	76.38

	With watermark			
Video sequence	PSNR(Y) (dB)	Total bits bits	Total encoding time (secs)	Bitrate (kbps)
Foreman	35.03	643888	1083.06	121.84
Coastguard	34.17	1144000	1133.36	215.76
Hall	36.28	249438	1034.05	48.08
Container	35.24	175896	1058.71	34.41
Carphone	37.08	407376	1057.51	77.52

The fragility of the watermark was tested under a variety of attacks such as transcoding, rotation, median filtering and cropping. The attacker would ideally want to attack the watermarked video in such a way such that the watermark is completely destroyed but at the same time there is no perceivable degradation in the quality of the video. The parameters of the attacks were chosen to create such a scenario. The performance of the fragile watermark under these attacks is shown in Table 5.2. The watermark detector at the decoder side was setup with the following parameters:

$$\text{If}(\rho_L = 0.5) < \Phi < (\rho_H = 1), \text{ then } \Phi := \eta$$
$$\text{else if } \Phi < (\rho_L < 0.5), \text{ then } \Phi := \zeta$$

where;

ρ_L: lower bound threshold of NC

ρ_H: higher bound threshold of NC

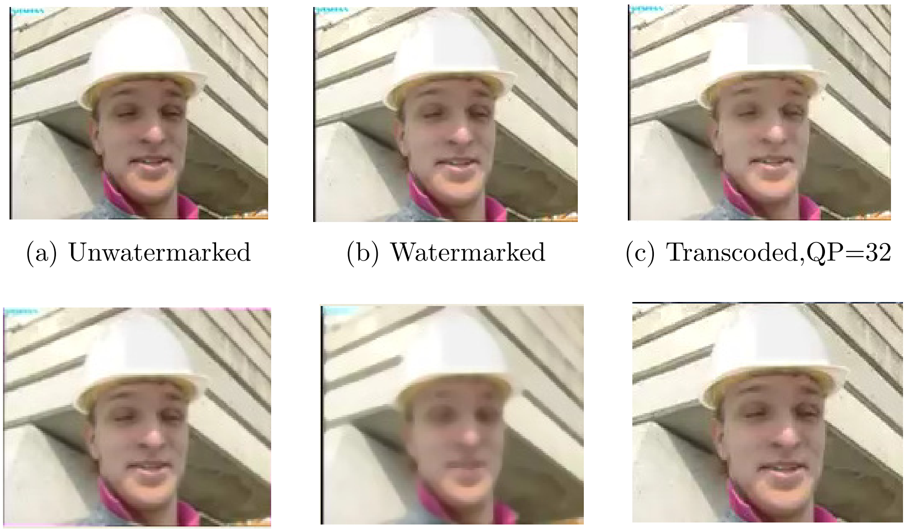
Φ: the detector response

η: the channel noise

ζ: tampering attack

Table 5.2: Watermark fragility evaluation under some common attacks

	Detector response Φ, NC ρ			
Video	Transcoding	Rotation	Median	Cropped
sequence	QP from 30 to 32	0.25^o	3×3 filtering	164×132
Foreman	0.53,0.53	0.55,0.53	0.45,0.45	0.47,0.47
Coastguard	0.45,0.43	0.46,0.40	0.47,0.46	0.49,0.47
Hall	0.36,0.30	0.46,0.42	0.47,0.45	0.61,0.58
Container	0.60,0.57	0.47,0.46	0.50,0.50	0.52,0.48
Carphone	0.41,0.41	0.43,0.46,	0.50,0.52	0.48,0.45



(a) Unwatermarked

(b) Watermarked

(c) Transcoded,QP=32

(d) Rotated 0.25^o

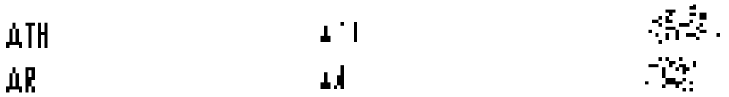
(e) Median 3×3 filtering

(f) Cropped to 164×132

Figure 5.2: Snapshots of an I-frame under some common attacks

Table 5.2 shows that $\Phi < 0.5$ for most of the attacks thereby verifying that the watermark is fragile and easily detects tampering. Setting a threshold level of $\Phi = 0.5$ is quite lenient considering that the watermark will be totally unrecognizable if more than half of its information is lost. Practically speaking, the threshold for Φ should be set at 0.75, below which the alarm of tampering could be raised. The justification for making such a statement is that only a well planned attack can distort the watermark beyond 25% and it is very rare that the channel noise would distort the mark beyond this threshold. Snapshots of the first I-frame of an attacked video with respect to a watermarked frame that has not been attacked are shown in Fig.5.2.

The watermark extracted from an un-attacked video sample is depicted in Fig.5.3a while Fig.5.3b and 5.3c show the watermark extracted after the transcoding and median filtering attack respectively. As can be seen, the watermark is destroyed under both the attacks. Similar observations were made for the rotation and cropping operations.



(a) Un-attacked (b) After transcoding (c) Median 3×3 filtering

Figure 5.3: Extracted watermarks

The rate-distortion (R-D) statistics and the R-D curve are depicted in Table 5.3 and Fig.5.4 respectively. The R-D measurements were taken with a QP of 24, 28, 32, 36 and 40. Table 5.3 only shows the observations for QP at 28 and 36 so as to compare with the results presented in [69]. In Table 5.3, Kuo *et al.* [69] and Qiu *et al.* [72] use H.264 encoded video as the host while Zhu *et al.* [73] and Zhang *et al.* [74] use MPEG2 as the host. It can be seen that the watermark payload for the proposed method is the highest except for Zhu *et al.* [73]. However, the video standard used in Zhu *et al.* [73] is MPEG2 which has much higher redundancy than the H.264 standard. This means that it offers more "space" to embed a much larger payload.

Figure 5.4 indicates that the proposed technique performs better than the other techniques, offering better R- D characteristics especially at higher bitrate values. Even though there is not much of a difference in the PSNR values among the techniques but the method proposed in this paper provides comparable PSNR values at a lower bitrate. This is so because the embedding technique only changes the value of *last_coeff* $(= c)$ by at most one 1 bit and that too very rarely. A manual inspection of the values of *last_coeff* while embedding the watermark in $Foreman$ revealed that only 8 values of c were modified for $n = 100$.

Table 5.3: Comparison of R-D characteristics

Sequence			*Foreman*		*Container*	
			QP			
Method	Payload (bits)	R-D	28	36	28	36
Unwatermarked		PSNR(dB)	36.89	31.30	36.22	30.72
		Bitrate(kbps)	84.11	28.97	32.44	8.81
Proposed	26×26,1bpp	PSNR(dB)	36.03	30.92	35.93	30.48
		Bitrate(kbps)	84.37	29.05	32.61	8.95
Kuo *et al.* [69]	not known	PSNR(dB)	35.76	30.60	35.86	30.39
		Bitrate(kbps)	85.46	29.21	33.01	9.32
Zhu *et al.* [73]	32×32,1bpp	PSNR(dB)	35.74	30.53	35.86	30.30
		Bitrate(kbps)	88.90	29.84	36.88	9.65
Zhang *et al.* [74]	64	PSNR(dB)	35.76	30.58	35.84	30.30
		Bitrate(kbps)	87.84	29.56	36.36	9.65
Qiu *et al.* [72]	99	PSNR(dB)	35.75	30.57	35.86	30.73
		Bitrate(kbps)	86.93	29.49	35.15	9.70

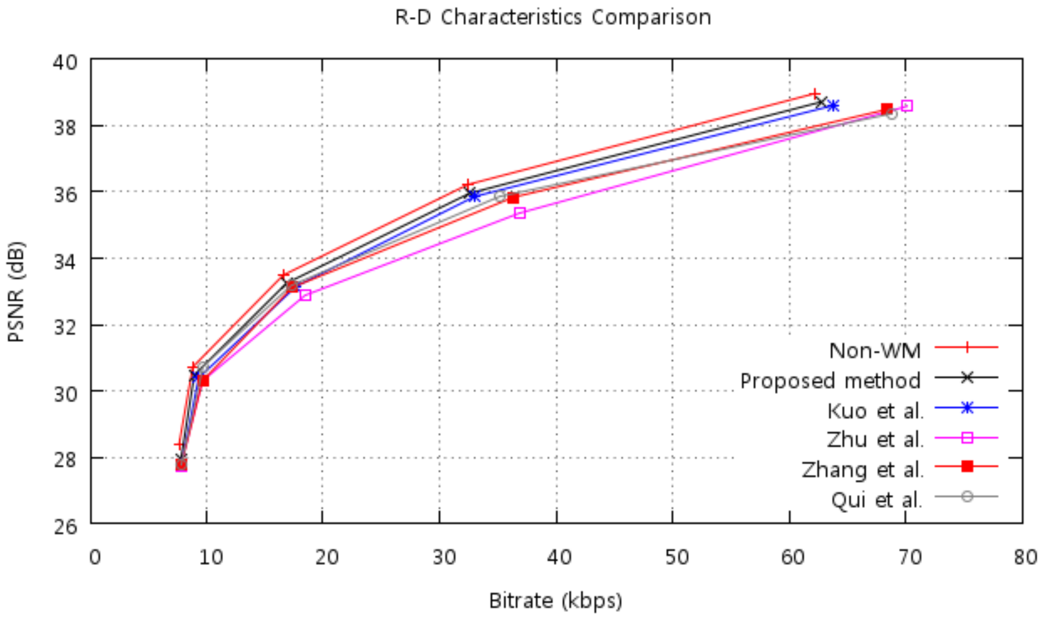


Figure 5.4: Rate-distortion curves for *Container*

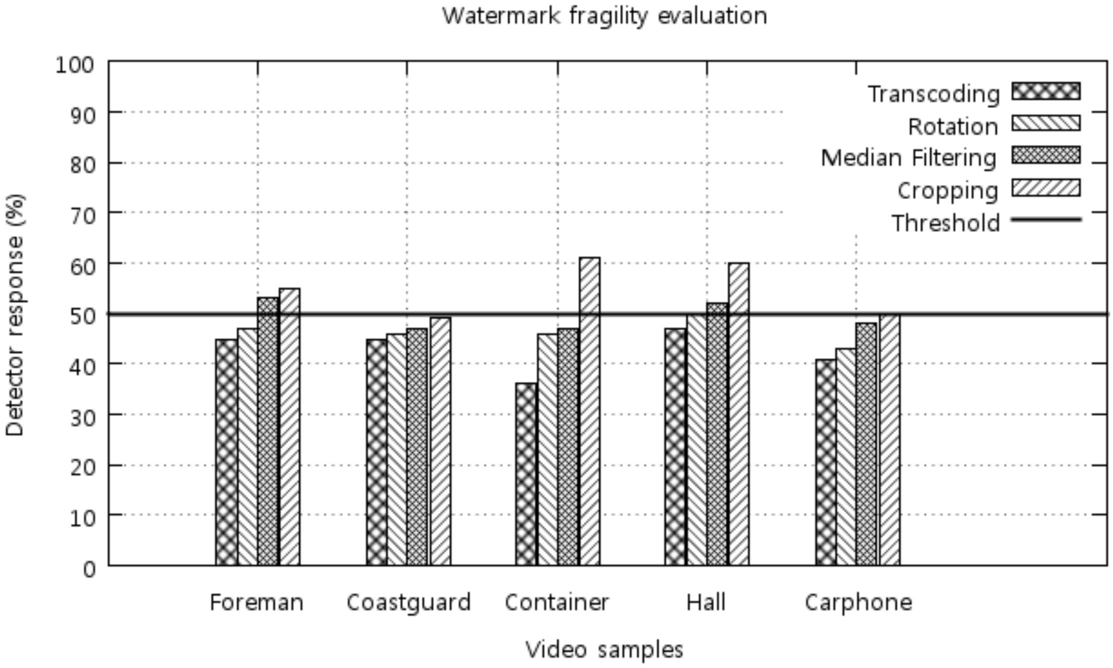


Figure 5.5: Watermark fragility evaluation

5.5 Conslusion

A fragile watermarking algorithm based on the CAVLC entropy method of H.264 AVC is proposed in this Chapter. The last (highest frequency) coefficient of a 4×4 quantized block within an I-frame is used to embed the watermark. The technique offers many advantages such as being blind and perceptually transparent while at the same time exhibiting encouraging R-D characteristics. The embedded watermark is fragile enough to detect most of the attacks. Figure 5.5 shows that the watermark gets perturbed even under the most common attacks and almost 50% of the watermark bits get altered which is significant enough to indicate tampering. Finally, the watermarking technique is computationally efficient and does not add any significant overhead to the H.264/AVC encoder and decoder.

The technique however, might fail in instances where an attacker might attempt to completely remove the watermark by dropping off most of the high frequency components from every 4×4 block. This could be achieved by setting a high quantization parameter. In such a scenario, the detector would fail to detect the presence of a watermark and hence tampering. However, the price the attacker would have to pay would be in terms of a resulting video having poor quality. The results of this methodology were published in [75].

Chapter 6

Reversible Fragile Watermarking

Watermarking digital content usually leads to irreversible changes in the host. This can be unacceptable in certain application areas due to a number of reasons, for instance, legal issues. To address these issues, the paradigm of reversible watermarking was developed. This chapter proposes a reversible watermarking technique based on the Difference Expansion (DE) technique proposed by Tian [76]. The watermark so embedded is fragile in nature and is useful for authentication systems. The proposed algorithm utilizes the IPCM macroblocks in an H.264/AVC bitstream to embed a high-capacity watermark. The performance of the proposed technique is evaluated for a variety of video samples and encoding parameters. The results show that the technique is capable of embedding a high payload in the H.264/AVC bitstream with a negligible effect on the rate-distortion characteristics while at the same time being completely reversible.

6.1 Introduction

As pointed out in section 3.5, embedding a watermark inevitably leads to distortions and introduction of unwanted artifacts. If the watermarking technique is irreversible, then these distortions are permanent. In certain application domains such as medical, surveillance and military, such irreversible distortions to the host is totally unacceptable. This may be due to legal reasons or due to the high level of sensitivity associated with the host data. In such application domains, irreversible watermarking techniques are clearly unsuitable. As a result, they require a reversible watermarking approach wherein the host is restored to its original undistorted form after the watermark has been detected/extracted.

A reversible watermarking system can also be effectively employed to authenticate streaming videos or its source in application areas such as video conferencing, Video-on-Demand (VoD) etc. in order to prevent counterfeiting. The

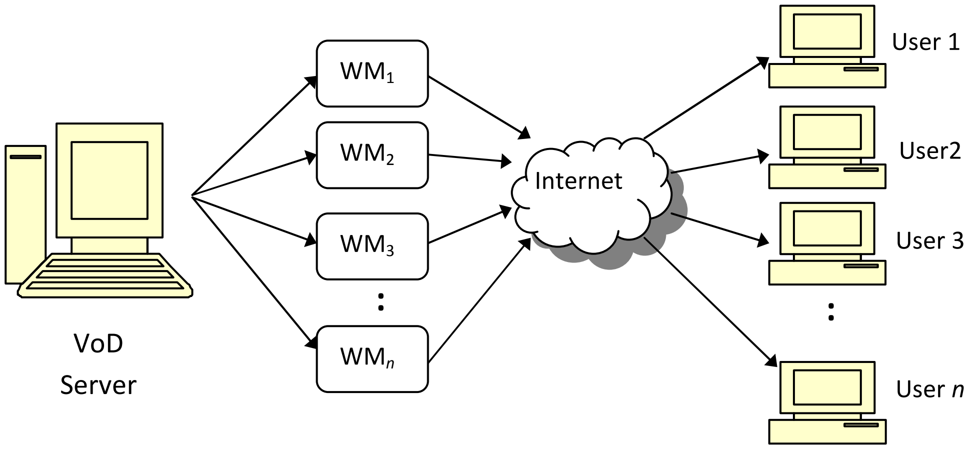
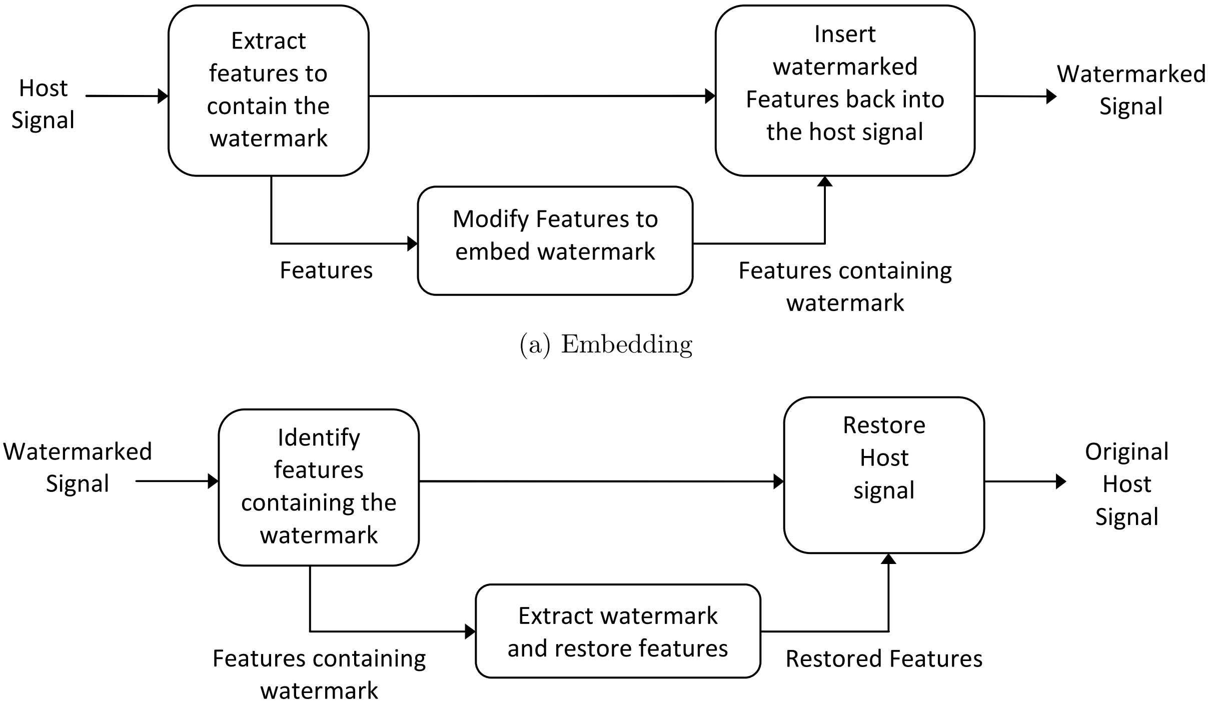


Figure 6.1: Watermarking in a multicast environment

advantage of reversible watermarking is that the quality of the video is not compromised since it is restored back to its original form before playback. In a VoD scenario, as depicted in Fig.6.1, the video content to be streamed can be reversibly watermarked at the server side with recipient specific information such as the International Mobile Equipment Identity (IMEI) number of a mobile device or the identification number of the video codec chip on the recipient's device. At the receiver side, each recipient will receive a watermarked video stream with a unique identification code as the watermark. Since the end-user has to be registered using a unique identification with the VoD server, the presence of that unique identification as a watermark in the received video stream will authenticate the source as legitimate. In addition, if the user receives a video stream with a tampered/damaged watermark, then it would indicate forgery or a counterfeiting attempt.

Reversible or lossless watermarking was first proposed by Hosinger *et al.* [77] in a patent owned by The Eastman Kodak, followed by a number of algorithms such as the Patchwork algorithm [78], its improvement proposed by Fridich *et al.* [79] and the patchwork histogram rotation algorithm [80]. Consequently, many other reversible algorithms were proposed for images. Celik *et al.* [81] proposed a generalized LSB data embedding algorithm that utilized the quantization residues to carry the payload. Lossless image compression algorithm was employed to compress the residual values to achieve high embedding capacity. Kalker and Willems [82] analyzed and proposed the theoretical limit of embedding a payload in an image using lossless data compression.

Reversible watermarking techniques can be broadly classified into two categories: those that involve additive spread spectrum techniques and those that modify the host signal. The first category [77, 83] involves superimposition of a spread spectrum signal corresponding to the watermark over the host signal. At the decoder, the watermark signal is detected and then the watermark is subtracted from the host signal to restore it back to its original form. The second category of techniques [79, 84–86] modify some portion of the host signal. This could be in the spatial, transform or the compressed domain. However, since the modification is mostly irreversible, information about those features that have been modified are also compressed and made part of the payload. This information is then used at the receiving end to restore the host signal back to its original form after extraction of the watermark. The reversible watermarking technique proposed in this paper belongs to the second category. Fig.6.2 depicts the general model of a reversible watermarking system.


Host Signal
Extract features to contain the watermark
Insert watermarked Features back into the host signal
Watermarked Signal
Features
Modify Features to embed watermark
Features containing watermark
Watermarked Signal
Identify features containing the watermark
Restore Host signal
Original Host Signal
Features containing watermark
Extract watermark and restore features
Restored Features

(a) Embedding

(b) Extraction

Figure 6.2: Reversible watermarking

It is well known that watermarking techniques designed for images can be modified to be applied to digital video as well. Fallahpour and Megías [87] proposed an error resilient reversible data hiding method for H.264/AVC wherein the prediction error values are modified and then used to embed data. They claimed to embed a large amount of payload while at the same time maintaining a high PSNR value. Kapotas and Skodras [88] proposed a real-time data hiding method that exploited the IPCM macroblocks to embed a watermark. The method used the 4 LSBs of every luminance and chrominance sample of an IPCM macroblock to embed a watermark. However, both of these techniques are irreversible in nature.

The proposed algorithm is also based on the idea of embedding watermarks within IPCM macroblocks but in a reversible way by employing the DE method [76]. This method is applied on a pair of 8-bit pixel values within IPCM macroblocks. In addition, the DE algorithm is also modified so that it is applied to only a single macroblock within P frames in contrast to being applied to the whole frame (image). The resulting algorithm is a reversible/lossless watermarking technique for H.264/AVC video.

6.2 Background

This section provides a brief overview of the difference expansion technique when applied to images and the IPCM macroblocks generated by the H.264/AVC encoder.

6.2.1 Difference Expansion

The DE method proposed by Tian [76] allows a high-capacity watermark payload to be embedded in images and video, while at the same time being completely reversible and maintaining a high visual quality. The technique works by selecting a pair of neighbouring pixels and embedding the watermark in the difference of their values. These difference values can be used to embed not only the message (watermark payload), but also the restoration information, message authentication code and any other secondary information.

Assume (x,y) are a pair of neighbouring pixel values, which could be adjacent pixels or pixels within a defined area of an image/frame such as a macroblock. Next, the difference h and average l are calculated between the pair. The difference value h is utilized to embed a watermark bit b. This is done by multiplying h by 2 (shift left by 1 bit) and appending b at the LSB position resulting in a new value h'. New (watermarked) pixel values (x', y') are then calculated using l and h' and sent across to the decoder. At the decoder side, (x',y') are used to extract the watermark bit b and are restored back to their original values (x,y). Shown below is an example of this technique.

Let x=206, y=212, b=1

Watermark embedding:

$$l = \left\lfloor \frac{x+y}{2} \right\rfloor = \left\lfloor \frac{206+212}{2} \right\rfloor = 209$$

$$h = x - y = 206 - 212 = -6$$

$$\left.h' = 2 \times h + b = 2 \times (-6) + 1 = -11 \qquad \right\} //\text{Embedding}$$

$$x' = l + \left\lfloor \frac{h'+1}{2} \right\rfloor = 209 + \left\lfloor \frac{-11+1}{2} \right\rfloor = 204$$

$$y' = l - \left\lfloor \frac{h'}{2} \right\rfloor = 209 - \left\lfloor \frac{-11}{2} \right\rfloor = 215$$

Watermark extraction:

$$l' = \left\lfloor \frac{x'+y'}{2} \right\rfloor = \left\lfloor \frac{204+215}{2} \right\rfloor = 209$$

$$h' = x' - y' = 204 - 215 = -11$$

$$\left.b' = LSB(h') = 1 \qquad \right\} //\text{Extraction}$$

$$h = \left\lfloor \frac{h'}{2} \right\rfloor = \left\lfloor \frac{-11}{2} \right\rfloor = -6$$

$$x = l' + \left\lfloor \frac{h+1}{2} \right\rfloor = 209 + \left\lfloor \frac{-6+1}{2} \right\rfloor = 206$$

$$y = l' - \left\lfloor \frac{h}{2} \right\rfloor = 209 - \left\lfloor \frac{-6}{2} \right\rfloor = 212$$

The above technique is a form of reversible integer transform called the Haar wavelet transform that can be applied to a pair of 8-bit pixel values. As can be seen there is a one-to-one correspondence between (x,y) and (l,h). However, it is necessary to ensure that the new (x',y') values do not overflow/underflow i.e. kept within the range [0, 255]. This can be represented as:

$$0 \leq l + \left\lfloor \frac{h'+1}{2} \right\rfloor \leq 255 \textbf{ and } 0 \leq l' - \left\lfloor \frac{h'}{2} \right\rfloor \leq 255$$

Since l and h are integers, the above condition can lead to:

$$|h'| \leq 2(255 - l) \textbf{ and } |h'| \leq 2l + 1$$

In fact, to prevent overflow and underflow, h' should satisfy:

$$|h'| \leq min[2(255 - l), 2l + 1]$$

Or, in other words, it can be formulated as:

$$|2 \times h + b| \leq min[2(255 - l), 2l + 1]$$

Values of h that satisfy the above condition are termed as *expandable* values. Thus only those pixel values will be chosen for DE whose h value is expandable. The hiding ability of h is $log_2 k$, where k is the largest integer such that:

$$|k \times h + b| \leq min[2(255 - l), 2l + 1], \quad \forall b, 0 \leq b \leq k - 1 \tag{6.1}$$

For this technique, the value of k is chosen to be 2 so the hiding ability of h is 1. The complete watermark embedding algorithm consists of the following stages: calculating difference values, creating a location map, data embedding using DE and inverse integer transform.These steps are discussed in detail in [76].

The DE algorithm can be applied to a frame more than once i.e. for an already embedded P frame, additional payload can be inserted within it but with a different pairing pattern. The second embedding doesnt perturb the previous embedding. This is termed as *multi-layer embedding*. In fact, one payload can be divided and spread across layers. It can be deduced that each layer has a payload capacity of less than 0.5bpp. This means that multiple layer embedding has a capacity of less than $M/2$bpp where M is the number of layers. Theoretically, multi-layer embedding could go on to any number of levels. However in practice, the redundancy keeps on reducing with increasing layers of embedding as more and more h values violate condition in equation 6.1. The payload size (i.e. the bit length) is thus limited only by the payload capacity limit.

6.2.2 Intra IPCM macroblocks

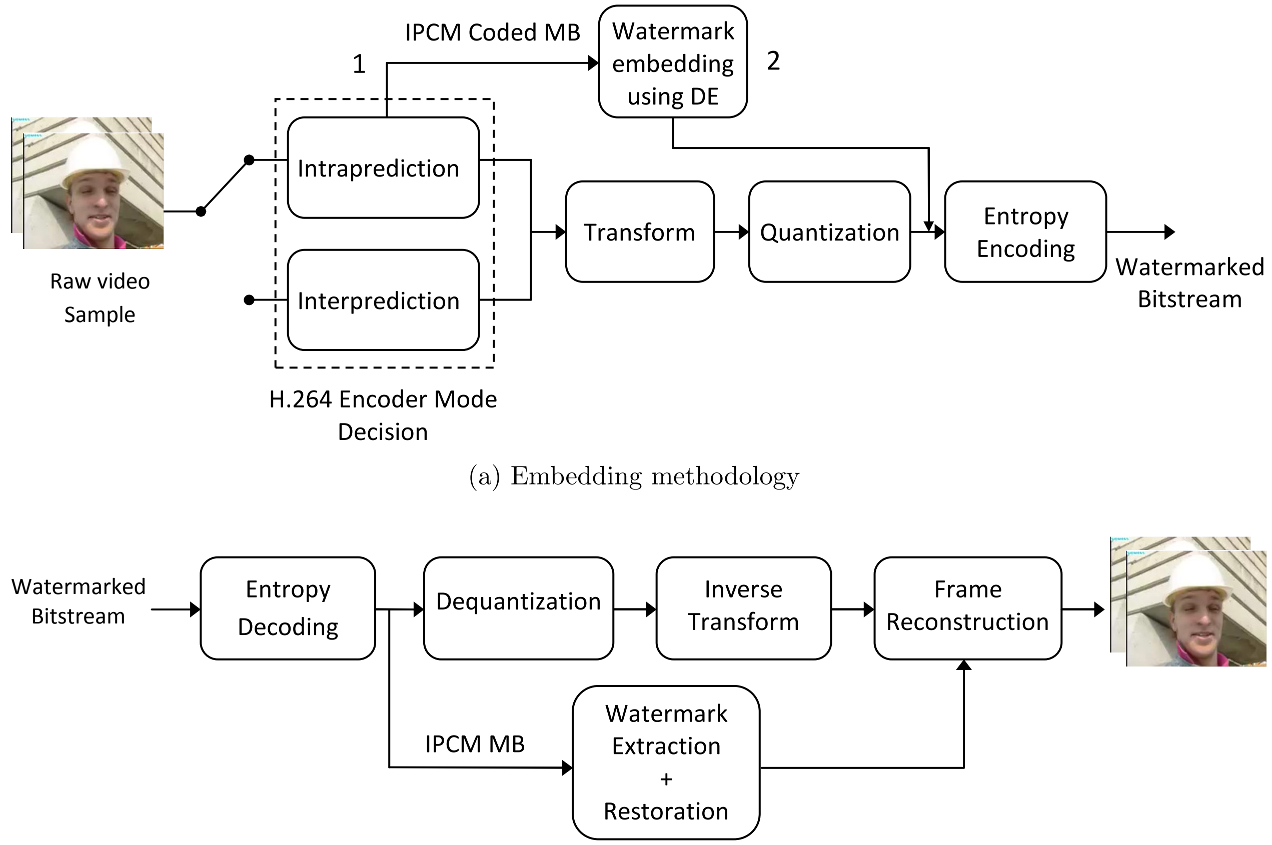
In certain rare and unusual situations, when the quantization step size is very small, thereby generating a high-quality coded video, the H.264 encoding process can actually lead to an increase in the overall number of bits required to represent the coded video when compared to the raw uncompressed video. Such a situation could arise in applications such as content archiving or distribution where a very high quality is required. Furthermore, it may be convenient for implementation reasons to have a reasonably-low identifiable limit on the number of bits necessary to process in a decoder in order to decode a single macroblock. These issues are taken care of by another type of intra coding method, called the IPCM macroblock mode. In this mode, the encoder decides to transmit the sample values without any prediction, transformation or quantization. In essence, it means that the samples are sent across as raw pixel values. This mode allows regions of the frame to be represented without any loss of fidelity. However, as pointed out in [89], this method is not efficient and neither was it meant to be. It was intended to be simple and to impose a minimum upper bound on the number of bits that can be used to represent a macroblock with sufficient accuracy. In fact, if one considers the bits necessary to indicate which mode has been selected for the macroblock, the use of the PCM mode actually results in a minor degree of data expansion.

From the above discussion, it is clear that the H.264/AVC encoder rarely generates an IPCM block. In fact, as part of their experiments, Kapotas and Skodras [88] were unable to obtain any IPCM macroblock naturally. Thus the encoder was forced to consider and encode specific macroblocks as IPCM macroblocks. The same strategy has been adopted for the proposed technique as is discussed in the next section.

6.3 Proposed Watermarking Technique

The H.264 encoder generates an encoded bitstream in the form of slices. Each slice in turn contains a slice header and a slice payload. Each slice payload contains within it a number of macroblocks. Each macroblock has a header and a payload. The header contains information such as prediction mode used to predict the macroblock. The macroblock payload contains the coded transform coefficients corresponding to the residual image samples after prediction. However, if the macroblock has been coded as an IPCM macroblock, then the macroblock payload contains raw pixel values. The DE watermarking method can only be applied to pixel values and not to residual coefficients, hence the justification for generating IPCM macroblocks.

As mentioned in the previous section, the watermarking technique involves forcibly generating IPCM macroblocks. Thus two steps are added within the H.264/AVC encoder (labelled as ‘1’ and ‘2’ in Fig.6.3a).



(a) Embedding methodology

(b) Extraction methodology

Figure 6.3: Proposed algorithm

The first step of the proposed method is 'IPCM macroblock generation', followed by 'watermark embedding'. In the first step, the encoders mode decision process is modified to force it to generate IPCM macroblocks. Normally, macroblocks towards the edge of a frame have a lesser degree of motion and detail associated with them in contrast to macroblocks located towards the centre of a frame. Therefore, for the proposed technique, it was decided to generate one IPCM macroblock in the top-left corner of each P frame (refer to Fig.6.4). Generating one IPCM macroblock per frame also kept the increase in the total number of bits to a minimum. If however, the encoder naturally generates an IPCM macroblock, then even that macroblock is utilized to embed the watermark information. A heuristic method could also be used within this step to make a more informed decision as to which macroblock should be generated as an IPCM macroblock. This decision could be taken on the basis of texture intensity or the amount of motion but at the expense of additional complexity. It should be noted that I- and B-frames are not used. Not modifying the I-frames lessens the extent of spatio-temporal error propagation since they are more frequently used for inter prediction than P-frames.

The second step of 'watermark embedding' involves accessing the raw pixel values within the IPCM macroblock generated in the first step and then embedding a watermark bit within a pair of pixel values on a multi-layer basis using the DE method. For the results in this article, two-layer embedding was attempted. In the first layer, the pairing is done vertically followed by horizontal pairing in the second layer (in fact, a key-based pattern can be used for added security but at the expense of additional computational overhead). Next, a 1-bit location map is created which indicates the selected expandable difference values. A value of '1' in the map indicates that h is expandable and '0' otherwise. The size of the one-bit bitmap is equal to the number of pairs of pixel values. The one-bit bitmap is then losslessly compressed using run-length coding. In addition, to indicate to the decoder that multi-layer embedding has been performed, a 16-bit header information is generated. Finally, the header information, the location map, and the watermark are combined together to form a payload and embedded into the video stream as per the procedure outlined in [76].

At the decoder side, the IPCM macroblock within a frame is identified and decoded. The location map is read and used to identify those pairs of pixel values that have been expanded for embedding. The payload bits are extracted out of the expanded pixel values which are then restored back to their original values.

The proposed algorithm utilizes both, the luminance and the chrominance components within an IPCM macroblock to embed the watermark bits. Theoretically, for an IPCM macroblock with 16×6 luminance (Y) components and 2(8×8) chrominance (U,V) components, pairing the pixel values would offer a maximum of [128 (Y) + 32 (U) + 32(V)] = 192 expandable h values per frame for the first layer. As explained in the previous section, for each successive layer of embedding, the number of possible expandable h values becomes progressively lesser than the maximum capacity i.e. less than 192. The payload capacity is thus limited by the total number of expandable h values within all P-slices (frames) of a video sequence. In general, for a video sequence with 'F' P-frames, the theoretical maximum available locations 'C' to carry the payload can be depicted as:

$$C = \sum_{i=1}^{F} n_i$$

where for a two-layer embedding scheme, each $n_i = h_{i1} + h_{i2}$, with h_{i1} and h_{i2} being the number of possible expandable h values in layer 1 and 2 respectively within frame i and $h_{i1} > h_{i2}$. Fig.6.4 shows a simplified schematic of the proposed watermark embedding system. The system at the decoder/receiver will follow the exact reverse sequence of steps in order to extract the watermark and restore the video.

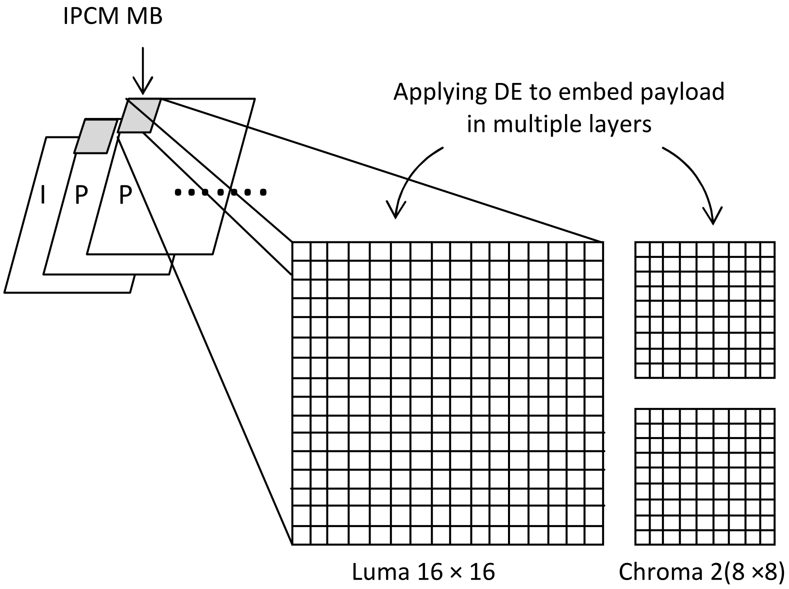


Figure 6.4: Overall system design

6.4 Experimental Results

The two steps outlined in the previous section were implemented within JM.15.1 reference software [62] of the H.264 Reference Software. Standard raw QCIF YUV(4:2:0) video samples, namely *Akiyo*,*Carphone*,*Foreman* and *Mobile* were chosen to be encoded and embedded with a watermark; however, only the results corresponding to *Foreman* and *Mobile* are reported here since other video samples returned similar results. The configuration parameter set for the encoder is listed in Table 6.1.

Table 6.1: Encoder configuration parameters

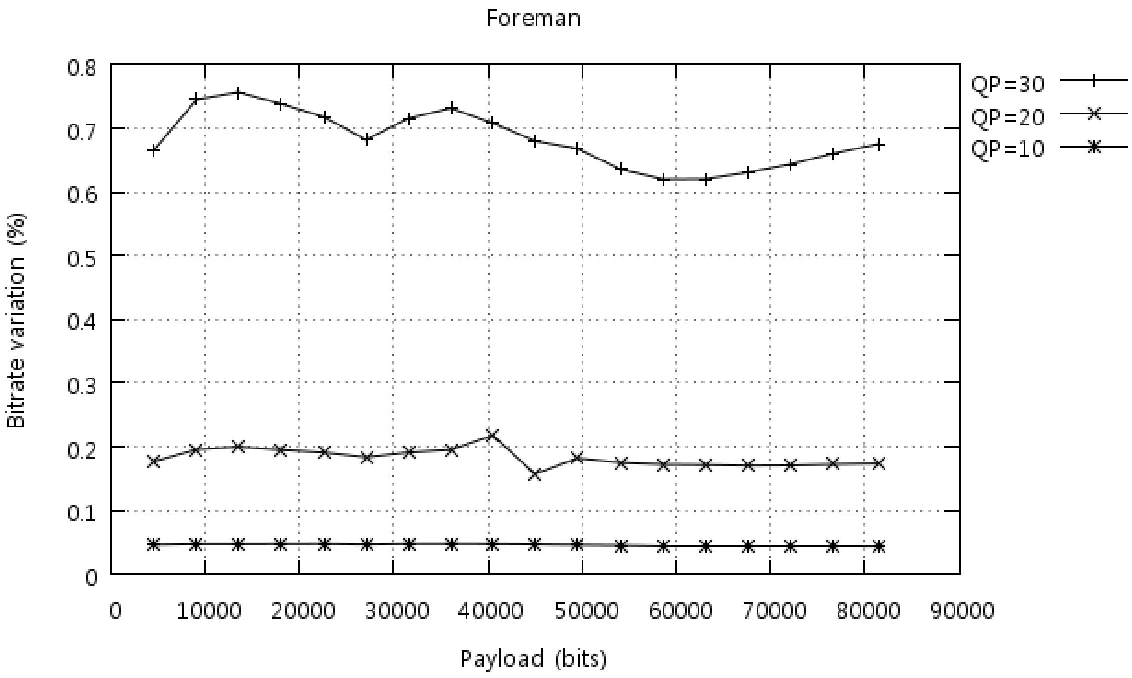
Number of frames to encode	300
Frame rate	30 frames per second
Frame resolution	176×144, 4:2:0
IDR period	16
Number of B frames	Not used
Entropy encoding	CABAC
RD optimization	High complexity mode

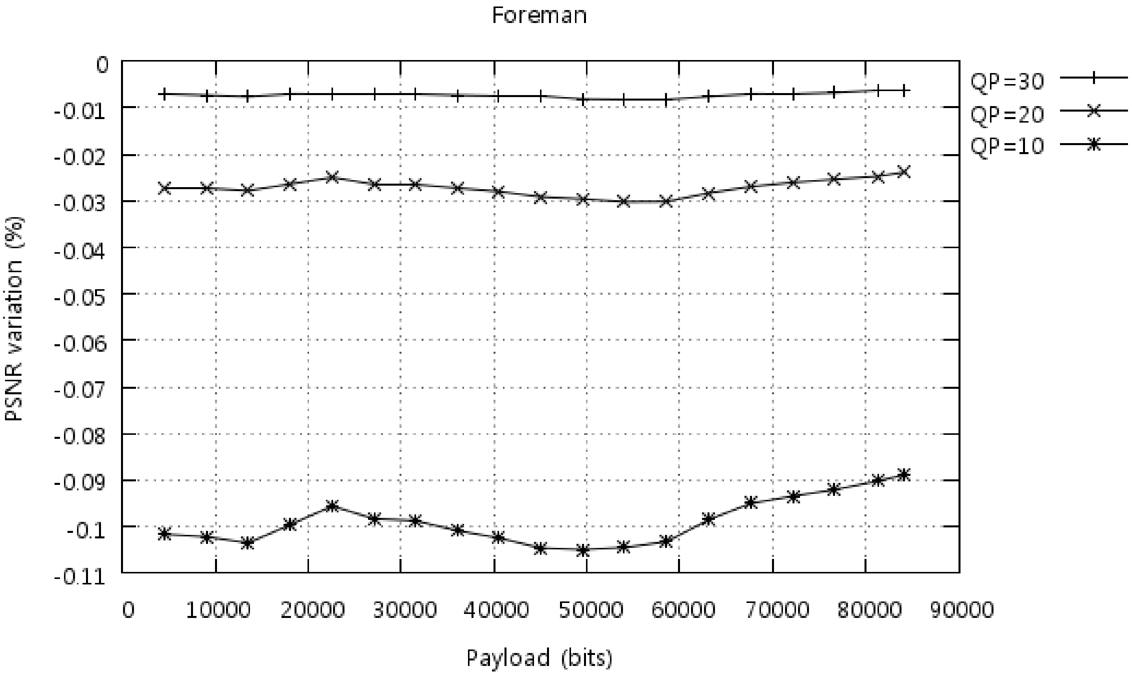
The frame encoding sequence was chosen to be *IPPP*.... B frames were not generated since the proposed technique only utilized IPCM macroblocks within

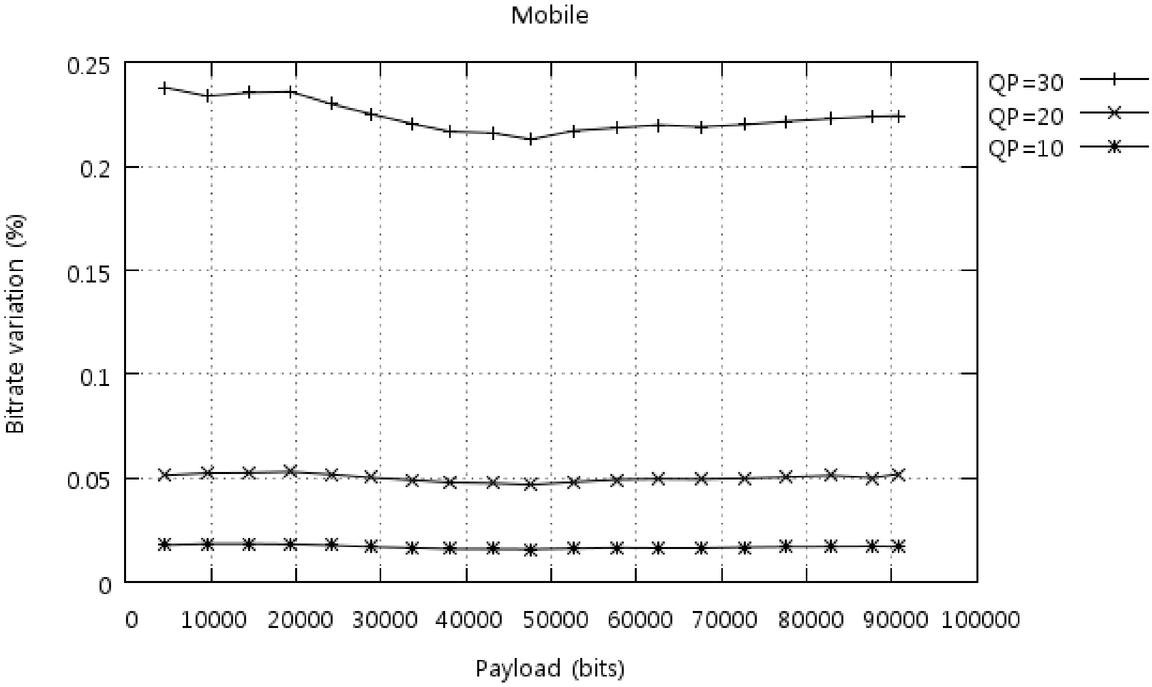
P slices to carry the watermark. The GOP length was fixed at 16 which implied that every 16th frame acted as an IDR boundary.

The performance of the proposed technique was measured with the rate control switched OFF and then with the rate control switched ON. Under rate control OFF, the value of the quantization parameter (QP) controls the quality of the video stream generated and transmission constraints are not considered. For the experiments reported in this article, the QP was set at 30, 20 and 10. When the rate control is turned ON, the encoder dynamically adjusts the QP of the video sequence being encoded depending upon the constraint limit set. This mechanism is useful when transmitting H.264 encoded video over a channel that has a limited bandwidth. For the experiments in this category, the encoder constraints were set at 60kbps, 50kbps and 40kbps. These are considered to be typical bitrates when transmitting video content over the internet. The graphs for the bitrate and PSNR variation with the rate control OFF and then for the rate control ON are shown in Fig.6.5 and 6.6 respectively.

From the results shown in the rate control OFF category, it can be seen that there is a negligible variation in both, the bitrate and PSNR values under increasing payload. The range of variation for the bitrate and the PSNR are depicted in Table 6.2 which indicates that embedding such a high payload does not have any significant impact on the R-D characteristics. A comparison is also made with the technique reported in [88] and is depicted in Table 6.3. It can be seen that the maximum variation of bitrate and PSNR within the proposed technique is much lower than the technique reported in [88].




Foreman
PSNR variation (%)
Payload (bits)
0
-0.01
-0.02
-0.03
-0.04
-0.05
-0.06
-0.07
-0.08
-0.09
-0.1
-0.11
0
10000
20000
30000
40000
50000
60000
70000
80000
90000
QP=30
QP=20
QP=10


Mobile
Bitrate variation (%)
Payload (bits)
0.25
0.2
0.15
0.1
0.05
0
0
10000
20000
30000
40000
50000
60000
70000
80000
90000
100000
QP=30
QP=20
QP=10

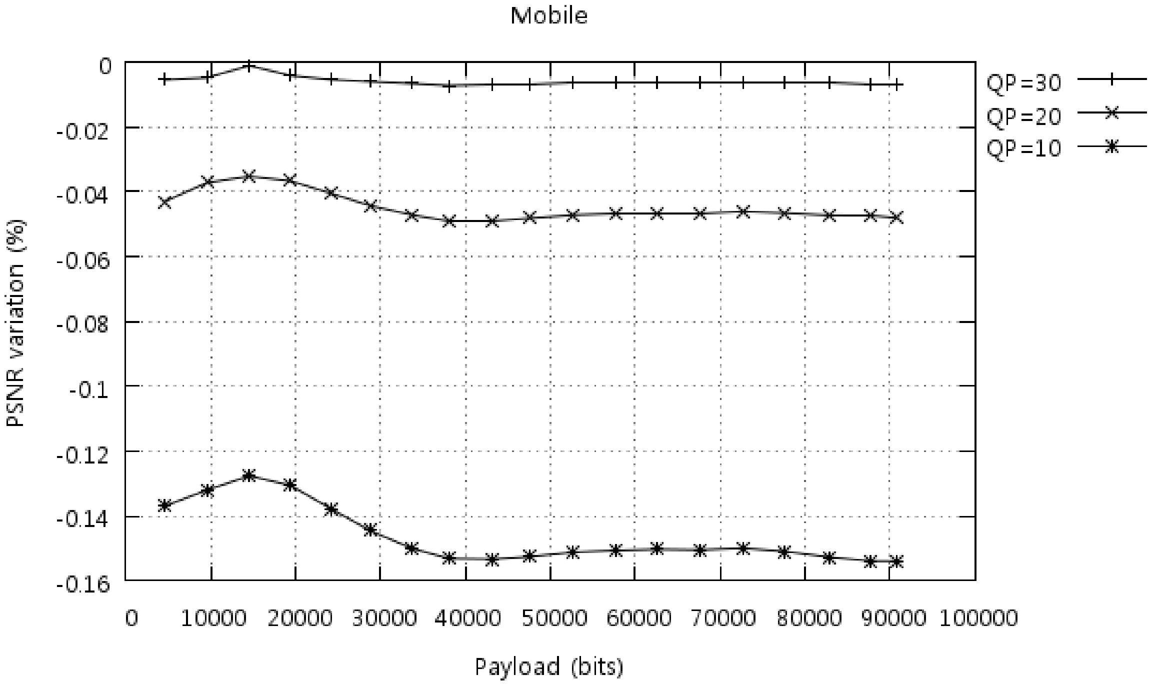


Figure 6.5: Bitrate and PSNR variation with rate control OFF

Table 6.2: R-D characteristics with rate control OFF

Video sequence	Range of bitrate variation (%)			Range of PSNR variation (%)		
	QP=30	QP=20	QP=10	QP=30	QP=20	QP=10
Foreman	0.619 to 0.755	0.157 to 0.217	0.043 to 0.047	−0.006 to −0.008	−0.023 to −0.030	−0.088 to −0.104
Mobile	0.213 to 0.237	0.047 to 0.053	0.015 to 0.018	−0.003 to −0.006	−0.035 to −0.048	−0.127 to −0.153

The above observations show that the proposed technique performs better than the one reported in [88] when it comes to bitrate variation while at the same time

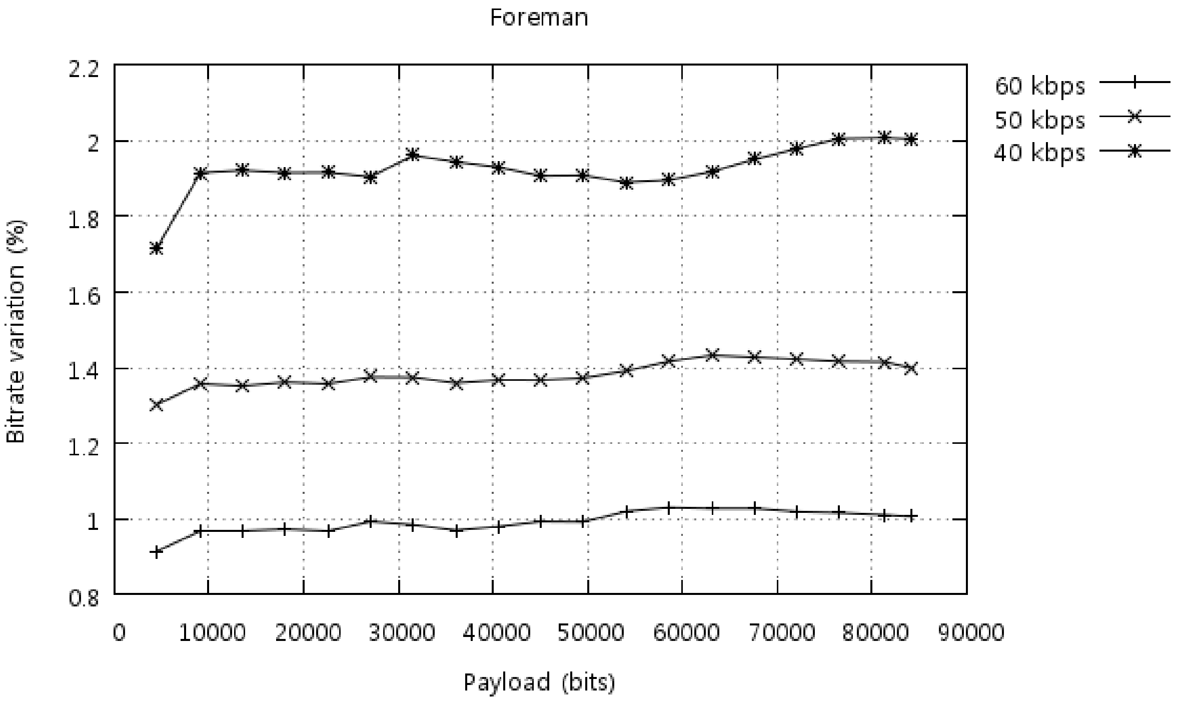
Table 6.3: Maximum variation characteristics with rate control OFF

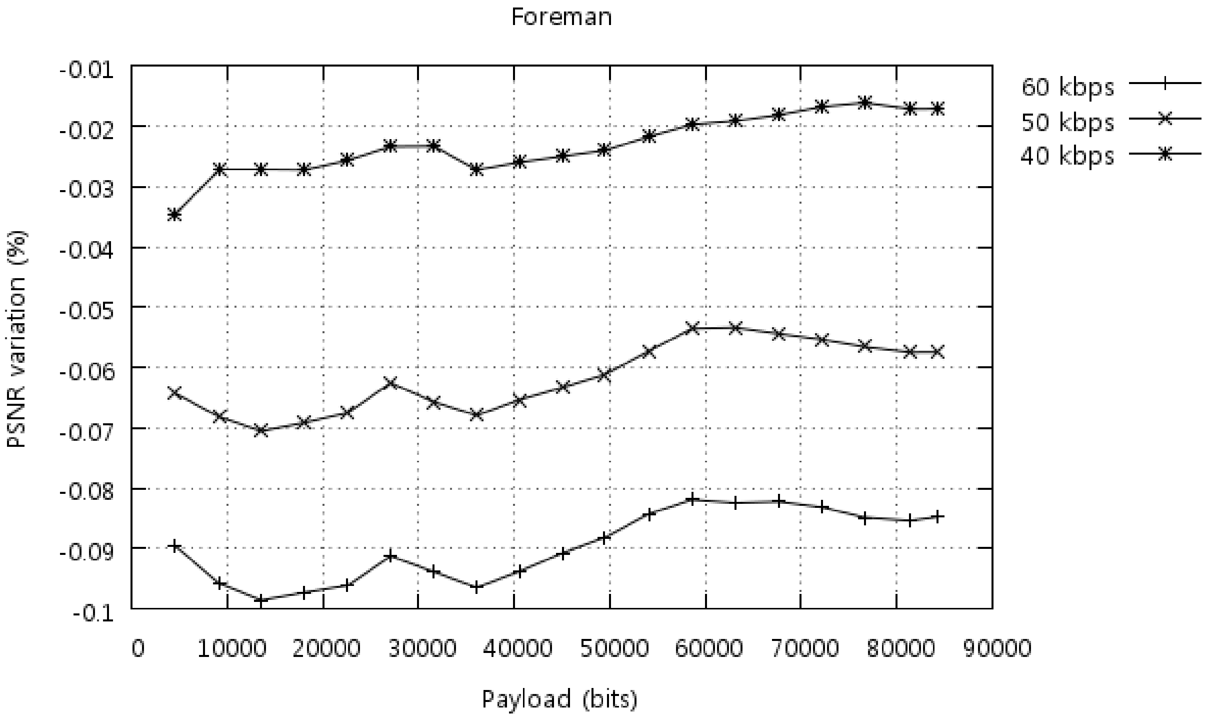
	Proposed method				Method proposed in [88]			
	QP=30		QP=20		QP=30		QP=20	
Video sequence	Bitrate (%)	PSNR (%)	Bitrate (%)	PSNR (%)	Bitrate (%)	PSNR (%)	Bitrate (%)	PSNR (%)
Foreman	0.75	–0.008	0.21	–0.03	3.5	–0.04	0.75	–0.2
Mobile	0.23	–0.006	0.05	–0.04	1.3	–0.01	0.14	–0.09

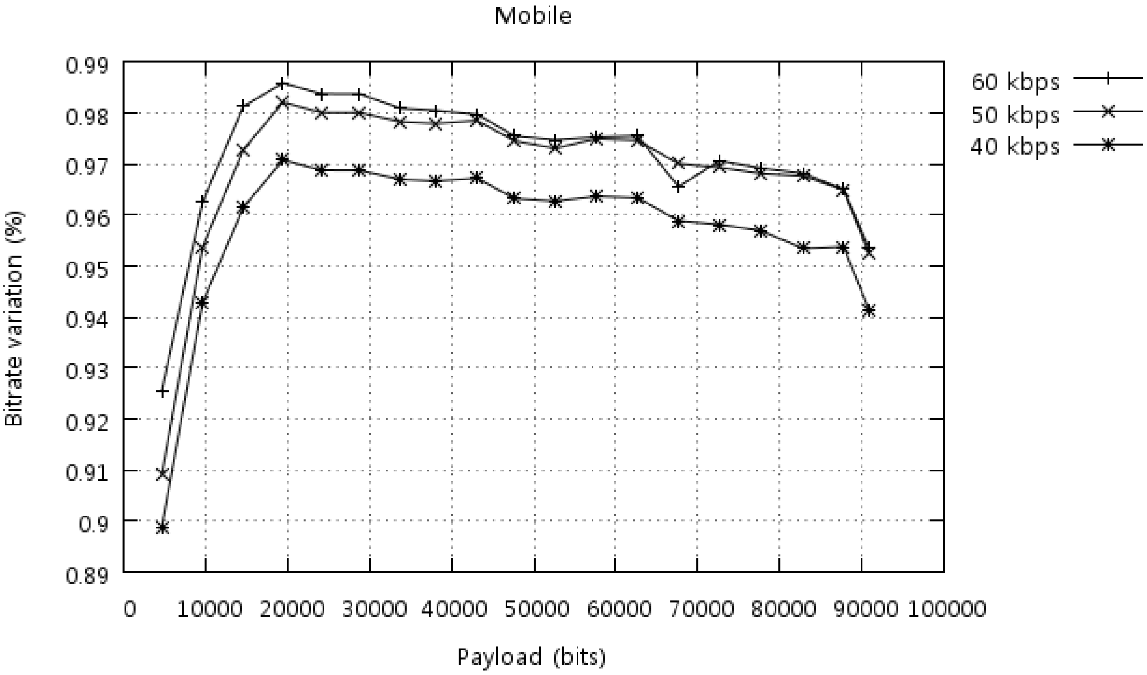
offering almost similar characteristics when it comes to PSNR variation.

The technique also exhibited a similar smooth performance with the rate control turned ON. It can however, be seen from Fig.6.6 and Table 6.4 that the range of variation, both for PSNR and bitrate, is higher than with the rate control turned OFF. In particular, *Mobile* exhibits a rather broad variation. However, the apparent wide variation in the bitrate and PSNR values in this case is due to the fact that *Mobile* is a high detail video sequence with a good amount of motion and forcing the encoder to generate an IPCM macroblock (under the watermarking technique) leads to an appreciable increase in the bitrate. Another reason for the sudden wide variation depends on the embedding capacity of specific frame/frames. This implies that higher is the payload embedded within a frame, higher will be the bitrate and PSNR variation exhibited at that point within the embedding process. For example, looking at the bitrate variation for *Foreman* in Fig.6.6, it can be seen that there is a sudden jump at around the 10,000 bit payload mark. This implies that the frame at this point in the video sequence was embedded with a large number of watermark bits since possibly all pixel-pairs were valid candidates to carry the watermark payload.

With the rate control ON, the proposed technique was again compared with the technique in [88]. The results are depicted in Table 6.5. It can be seen that maximum reduction in the PSNR for the proposed technique is relatively lower than the technique proposed in [88], although not by a significant margin. However, in agreement with the technique reported in [88], the cost of obtaining a much smoother performance within the proposed technique has been put on the PSNR. The proposed technique, however, offers a smoother performance at very low bitrates. For instance, at 40kbps where the degradation in the PSNR value is 0.26dB at the most(see values for *Mobile* in Table 6.4). This is lesser than the figure in [88] which reported a degradation factor of 0.43dB even at a higher bitrate of 60kbps.


Foreman
Bitrate variation (%)
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0
10000
20000
30000
40000
50000
60000
70000
80000
90000
Payload (bits)
60 kbps
50 kbps
40 kbps


Foreman
PSNR variation (%)
-0.01
-0.02
-0.03
-0.04
-0.05
-0.06
-0.07
-0.08
-0.09
-0.1
0
10000
20000
30000
40000
50000
60000
70000
80000
90000
Payload (bits)
60 kbps
50 kbps
40 kbps



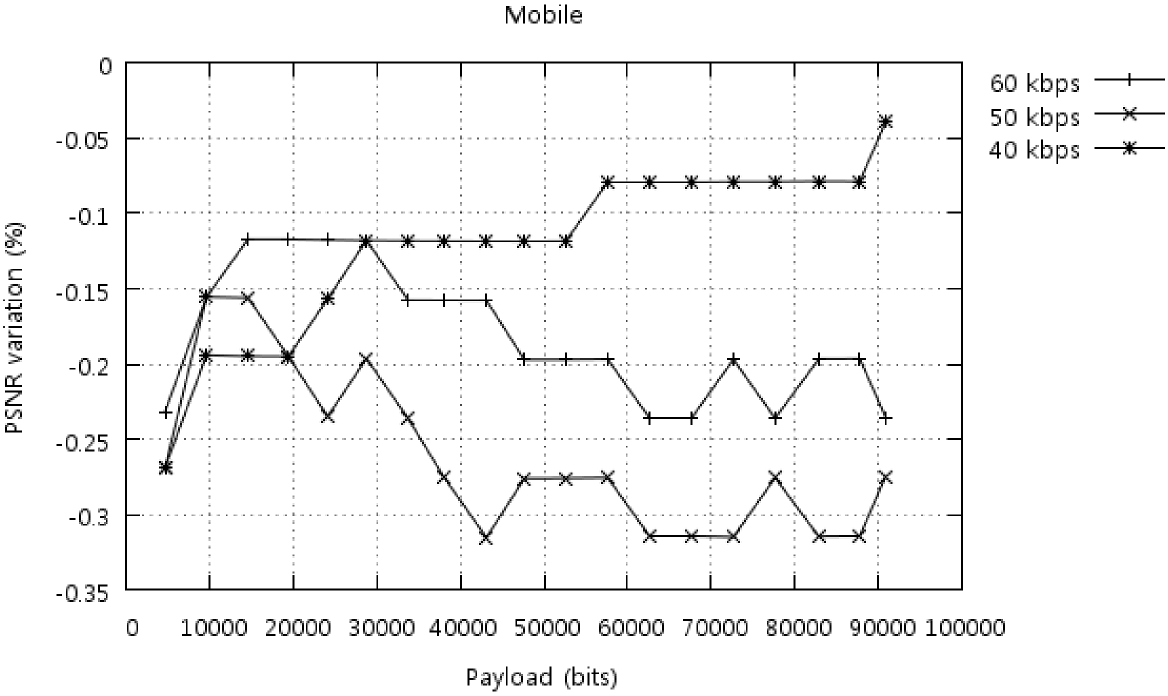


Figure 6.6: Bitrate and PSNR variation with rate control ON

Table 6.4: R-D characteristics with rate control ON

Video sequence	Range of bitrate variation (%)			Range of PSNR variation (%)		
	60kbps	50kbps	40kbps	60kbps	50kbps	40kbps
Foreman	0.914 to 1.029	1.302 to 1.432	1.716 to 2.008	−0.081 to −0.097	−0.053 to −0.070	−0.017 to −0.034
Mobile	0.925 to 0.985	0.909 to 0.982	0.898 to 0.970	−0.117 to −0.236	−0.155 to −0.314	−0.039 to −0.269

Table 6.5: Maximum variation characteristics with rate control ON

Video sequence	Proposed method				Method proposed in [88]			
	60kbps		50kbps		60kbps		50kbps	
	Bitrate (%)	PSNR (%)	Bitrate (%)	PSNR (%)	Bitrate (%)	PSNR (%)	Bitrate (%)	PSNR (%)
Foreman	1.02	–0.09	1.43	–0.07	N/A	–0.24	N/A	–0.36
Mobile	0.98	–0.23	0.98	–0.31	N/A	–0.23	N/A	–0.34

Next, the payload capacity was compared to the values reported in [88] and [90]. In order to perform a fair comparison, the encoder was set to exactly the same configuration as that mentioned in [88]. The results are shown in Table 6.6 which prove that the proposed technique offers a much higher payload capacity when compared to the values reported in [88] and [90]. In fact, embedding one bit per h under a two layer embedding scheme, the proposed technique is capable of hiding, on an average, 159.01 bits per QCIF frame in contrast to 61.4 bits per QCIF frame as reported in [88]. The payload values for the proposed technique contain within them not only the actual watermark bits but also the compressed location map and the header information. The average size of the compressed location map for this set of experiments was found to be 21.7 bits per IPCM macroblock/frame for a two-layer embedding. Thus, the average size of the location map to be embedded for a 300 frame video sequence (for instance, *Grandma*) is 6097.7 bits. This value combined with a 16-bit header information (indicating multi-layer watermarking) leaves out an actual watermark capacity of $[48000-(6097.7+16)] = 41886.3$ bits on an average, which is significantly higher than the capacities reported for the other two techniques. It should be noted that the maximum theoretical payload capacity of 48000 bits is obtained when B frames are present within the video sequence and which were not considered as candidates for watermark embedding. Having a video sequence consisting of I- and P-frames only would obviously allow a much higher embedding capacity, as is shown in Table 6.7.

Table 6.6: Comparison of payload capacity

	Payload (bits)			
Video sequence	Proposed technique		Method	Method
300 frames (IBPBP..)	Layer1	Layer2	in [88]	in [90]
Grandma	25344	48000	15360	12352
Bridge-close	25328	46288	15360	11748
News	25344	48000	18432	9972
Silent	25344	48000	18432	17368

Finally, the computational efficiency of the proposed algorithm was analyzed by taking into the account the time taken by the modified H.264/AVC encoder to encode a video sequence and embed the payload. The results are shown in Table 6.7. As can be seen, the increase in total encoding time is around 3% on an average, which makes the technique ideal for real time applications. The small computational overhead is due to the usage of simple mathematical operations for watermark embedding.

Table 6.7: Encoding time for the proposed technique

			One IPCM macroblock per P frame	
	Watermark payload		Total encoding time for 300 frames (secs)	
Video sequence	Layer1	Layer2	Un-watermarked	Watermarked (2 layers)
Akiyo	53671	89639	1450.1	1491.7
Carphone	53952	87052	1428.0	1470.2
Foreman	49855	84226	1622.7	1661.2
Mobile	52586	90879	1434.2	1495.4

6.5 Conclusion

In this chapter, a reversible watermarking technique was proposed which was based on the concept of difference expansion. The idea of embedding the watermark within the difference values of a pair of pixels had already been proved in [88] to offer the luxury of embedding a high payload. This concept was utilized within the H.264/AVC encoder by forcing it to generate a single IPCM macroblock within each P frame and applying DE on its pixel values. The performance of the proposed technique was tested on a variety of video sequences and was found to be capable of embedding a much higher payload than other similar reported algorithms. In addition, the proposed technique also exhibited encouraging R-D characteristics even at low birates. The embedded watermark being reversible, is fragile in nature which is a desirable trait when the requirement is to detect tampering within the video. Since the technique embeds the watermark in the spatial domain, it is resistant to common errors such as drift and other visual artifacts which are characteristic of compressed and transform domain techniques. Finally, the technique exhibits a very low computational complexity as the watermark embedding operation involves simple mathematical operations. This makes it ideal for hand-held devices and real-time application. Being a fragile watermarking method, the proposed technique would be vulnerable against the same category of attacks as mentioned in the previous chapter. However, improvements can be made by improving the choice of the macroblock to be encoded as an IPCM macroblock. A highly textured/detailed macroblock will always be encoded using low quantization parameters so as to retain most of its information. Choosing such a macroblock to be encoded as an IPCM macroblock and then embedding the payload within it would ensure that the watermark is not completely removed. But this would be at the expense of a higher resulting bitrate and added complexity. The results of the proposed watermarking technique were published in [91–93] and an extension of the method is under review at [94]

Chapter 7

Content Based Copy Detection

This Chapter introduces an efficient video copy detection method for the H.264/AVC standard. The mechanism is based on content based copy detection (CBCD). The proposed method divides each frame within a group of consecutive frames into a grid. Each corresponding grid across these groups of frames is then sorted in an ordinal vector which describes both, the spatial as well as the temporal variation. This ordinal matrix based copy-detection scheme is effective in not only detecting a copied video clip but also its location within a longer video sequence. The technique has been designed to work in the compressed domain which makes it computationally very efficient and hence suitable for a large class of application domains. The proposed mechanism was tested on a number of video sequences containing copies which had undergone a variety of modifications. The results prove that the proposed technique is capable of detecting these copies.

7.1 Introduction

Ordinal measures have proven to be the best method for matching video sequences in order to detect copies [103]. As with any video sequence matching method, this approach also has its roots in image matching methodologies. The most straightforward approach is by applying ordinal measures to every frame within a video sequence and then matching the resulting ordinal matrix fame-by-frame. However, this method is clearly inefficient since it fails to exploit the temporal characteristics of a video sequence.

Video can be considered to be a sequence of activity over a period of time. Each frame within a video sequence captures a part of that activity. These frames have a specific temporal sequence in order to depict the complete trajectory of that activity. Therefore, in order to design an effective video matching system, two factors have to be considered: the temporal order of the frames and the length of the video sequence. This means that in order to effectively detect a copy of a video sequence, spatio-temporal signatures should be utilized. Ordinal measures are quite capable of capturing and producing such a signature. Since temporal information is also captured, it offers the luxury of searching a video database for a specific kind of activity. For instance, in a database containing videos of football matches, queries can be made to search for all clips that show a goal being scored. Such a query cannot be made if the signature is solely based on spatial features.

CBCD methods have found acceptance in a number of applications such as detecting online copies of videos on torrents and media tracking. Media tracking involves detecting the usage of a specific piece of media in terms of its time, location and frequency. This method has found widespread acceptance in the marketing and advertising sector especially for TV broadcasts, where a competitor's commercial can be tracked to obtain relevant information. The tracking results can be used for copyright management, claim unfair practices or royalty payments. Another interesting usage of CBCD methods is to improve the search results of multimedia search engines where copies of the searched-for digital media could be removed before displaying the search results thereby reducing redundancy.

In CBCD methods, usually a shorter query clip is matched to a longer target clip in order to detect if a copy of the former exists as a copy within the latter. If a copy is detected then it is desirable that the detection mechanism also identifies the location of the copy. This is usually termed as temporal localization [48]. Video copy detection mechanisms that are based on key-frame (image) matching methodology will obviously be unable to perform this step.

H.264 video uses a number of novel methods to encode not just the spatial features but also temporal ones. The system proposed in this chapter utilizes some of these novel methods to generate an ordinal spatio-temporal signature. The technique was designed with an initial premise that a copy detection mechanism for H.264 video based on these methods will provide a much better performance as compared to using any other generic feature such as luminance values.

7.2 Existing CBCD Systems

As mentioned in Section 3.6, CBCD methods usually extract a signature from the video sequence. The signature can be extracted from spatial features such as luminance, from colour information such as the histogram of pixel values, temporal features such as motion vectors or even a combination of these features. Using spatial and colour features only for copy detection can be treated as a problem of image signature matching since these methods do not take into account the temporal nature of the video. Conversely, techniques developed using only temporal characteristics are ineffective against simple image processing operations such changes in luminance, contrast, colour etc. The following paragraphs outline some of the CBCD methods in each of the above mentioned domains.

One of the earliest CBCD methods based on spatial features was proposed by Bhat and Nayar [48] wherein the ordinal measure of every frame in the clip was computed and then matched to detect copies. This was achieved by dividing each frame into N×N blocks and sorting the average intensities of each block within that frame to give a rank matrix. Detecting copies can then be considered to be a problem of matching the rank matrices frame-by-frame between the original and copied videos. Lee and Yoo [95] designed a video fingerprinting method based on the centroid of gradient orientations. This method was claimed to be resistant towards most of the common video processing steps such as resizing, compression, frame rate change etc. Other spatial techniques such as those based on differential luminance [96]and edge detection [97, 98] have also been proposed.

Colour based CBCD methods are usually based on generating a unique signature from the colour histogram. Lienhart, Kuhmunch and Effelsberg [99] proposed a method where the colour coherence vector was used to characterize key frames of a video clip. Sanchez, Binefa and Radeva [100] proposed the use of principal colour components within the histogram of key frames for copy detection.

A number of techniques based on the temporal nature of video sequences have also been proposed. Indyk, Iyengar and Shivkumar [101] proposed some of the earliest CBCD methods based on exploiting the temporal characteristics of video. They treated the time duration between shot transitions as a unique signature. Radhakrishnan and Bauer [102] used the frame difference method based on projections of difference images between consecutive video images to extract a robust signature. They claimed the method to be resistant towards signal processing operations such as changes in luminance, compression, resolution changes and scaling. Hampapur, Hyun and Bolle [103] designed a copy detection technique based on motion vectors.

However, early experiments proved that ordinal signature based CBCD systems offer the most promising results. In fact, it has been proved in [103] that ordinal measurements not only offer the best performance when it comes to detecting copies but also in terms of computational efficiency. Consequently, a number of techniques have been proposed over recent years that are based on the ordinal signature approach. Some of them are briefly outlined as follows. Kim [45] proposed an ordinal measure of DCT coefficients which was based on the relative ordering of AC magnitude values. This method offers better performance than the ordinal measure of intensity, which is unable to detect basic operations such as horizontal or vertical flipping of images. Kim and Vasudev [49] combined the spatial and temporal features of video to design a spatiotemporal sequence matching system. The system combined spatial matching of ordinal signatures and temporal matching of temporal signatures to detect copies. Chen and Stentiford [50] designed a CBCD system based on temporal ordinal measurements. Each frame was divided into a 2×2 grid and corresponding grids were sorted in an ordinal ranking sequence along a time series. This measurement captured both, the local and global description of temporal variation. Nie *et. al.* [51] partitioned the key frames into blocks and computed their ordinal measure. Then they evaluated its 64-point DCT and extracted a fingerprint out of it after discarding some of its components. This fingerprint was utilized to detect copies and it was claimed to be quite efficient in detecting copies in long video sequences.

Categorizing CBCD methods in terms of spatial, transform or compressed domain is another way of approaching the problem. Spatial [52,100]and transform domain [45, 51] techniques though computationally more expensive, are simpler and more straightforward in design. Conversely, compressed domain approaches require a more in-depth understanding of the resulting bitstream in order to design the system but are usually more efficient computationally. A couple of compressed domain approaches based on motion vectors are reported in [53,103] and another technique utilizing transform coded coefficients is presented in [54].

The idea proposed in this chapter is to design a compressed domain CBCD method based on the spatiotemporal ordinal measurement. This would combine the high performance of spatiotemporal based ordinal measures with the computational efficiency of compressed domain approaches. The algorithm is designed based on features and characteristics that are unique to the H.264/AVC codec.

7.3 Proposed CBCD System

It can be seen from Section 7.2, that most of the CBCD systems are either spatial or temporal-only systems. Spatial-only systems are inefficient when it comes to detecting copies of video sequences since they fail to capture the temporal characteristics such as the sense of motion. Temporal-only systems, on the other hand, would be clearly ineffective against simple changes to video sequences such as changes in colour,contrast,luminance etc. [49].

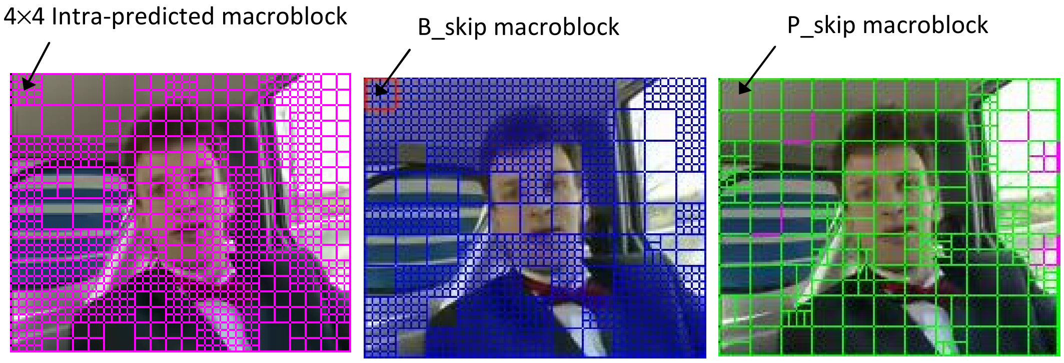
Thus, it is clear that a combination of spatial and temporal information offers the most efficient way to match videos.To realize a spatio-temporal based system for matching H.264 videos, it was imperative to identify those aspects of the H.264/AVC encoder that generate the relevant information and combine them together to generate unique signatures. Looking at the functioning of the encoder at different parameter settings, it was found that 4×4 intrapredicted macroblocks within the I-frames as well as P- and B-skip macroblocks within P- and B- frames respectively, were capable of retaining both, spatial and temporal information. We further argue that any modifications to the video might change the number of 4×4 intrapredicted, P- and B-skip macroblocks in each of the respective frames but the ordinal rank matrix built out of these frames would not be perturbed since the change would be consistent across these frames. Hence, they could be used as candidates to generate spatiotemporal signatures for a given video sequence. However, before going further with this discussion, we first briefly explain the concept behind the above mentioned macroblocks.

7.3.1 4×4 Intrapredicted Macroblocks

As explained in Section 2.2, intra-prediction [104] is a new technique that has been incorporated in the H.264 AVC video standard. In this mode, sample values of some macroblocks are predicted from neighboring macroblocks of the same frame. The predicted block is normally termed as prediction block P. For the luminance samples, P can be formed either using intra 4×4 or intra 16×16 mode. The intra 4×4 mode is used in the detailed and high motion areas of the frame, while the intra 16×16 mode is used in the smooth and the stationary areas of the frame. There are a total of 9 optional prediction modes for the intra 4×4 luminance block, 4 modes for 16×16 luminance block and 4 modes for the chrominance components. Fig.7.1a below illustrates an I-frame with most of the macroblocks being encoded using the intra 4×4 mode.

7.3.2 P_skip and B_skip Macroblocks

During inter-frame prediction within P- and B-frames, if the amount and degree of motion is quite low, then rather than using motion compensated macroblock modes, such macroblocks can be encoded as “skip type” macroblocks (see Section 2.3). For such type of macroblocks, neither a quantized prediction error signal, nor a motion vector or a reference index parameter is transmitted. Such “skipped” macroblocks are reconstructed by referencing the frame located at index 0 in the multipicture buffer (which is in fact, similar to how a P_16×16 macroblock would be reconstructed) [104]. P and B_skip macroblocks are very useful in encoding large areas within a frame that have no change or a slow constant motion like panning by reducing the actual number of bits that are transmitted. Fig.7.1b and 7.1c depict a B- and a P-frame with B_skip and P_skip macroblocks respectively.



(a) I-frame (b) B-frame (c) P-frame

Figure 7.1: Sample video frames encoded under the H.264 standard

7.3.3 Signature Design

Kim and Vasudev [49] showed that an ordinal matrix obtained by partitioning the frame into 2×2 regions is robust to most common modifications that can be done on a video sequence. In line with this argument, we begin by dividing the frames into 2×2 regions and representing them as TL (Top Left), TR (Top Right) ,BL (Bottom Left), BR(Bottom Right). We then count, in all the four regions, the number of 4×4 intra-predicted macroblocks within the I-frame, the number of B_skip macroblocks within the B-frame and the number of P_skip macroblocks within the P-frame. The resulting counts in each of the corresponding regions for each frame are then ranked as an ordinal vector along the time line. These vectors are then combined together to give the final ordinal matrix which acts as the signature. Fig.7.2 below explains how the signature is generated using a combination of three different frame types. Fig.7.2a depicts the division of frames into 2×2 regions. In order to ensure that the macroblock count in each region of the frame is an integer, it may be necessary to divide the frame into unequal-sized regions. However, we still attempt to keep the division as equal as possible. For instance, if the frame resolution is 176×144 then there would be 11 macroblocks along the x-axis and 9 macroblocks along the y-axis as illustrated in Fig.7.2a. Hence in this case, the division was made after the 5th macroblock along the x- and the y-axis, starting from the top-left corner macroblock. In general, the approach adopted is:

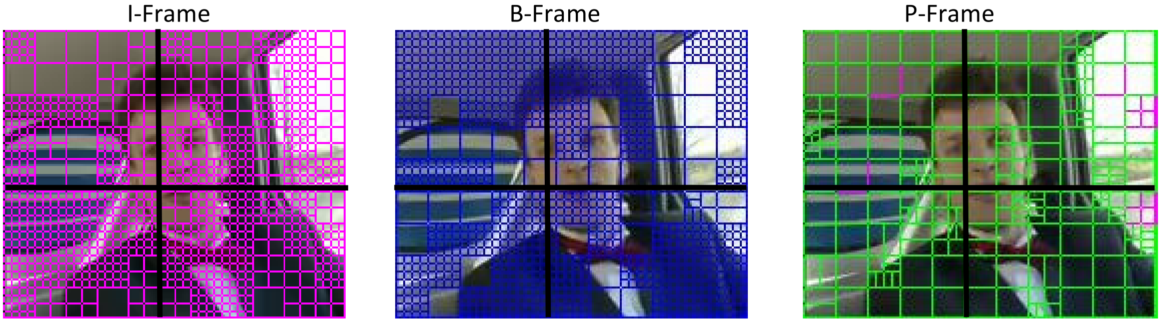
$$x_{partition} = \left\lfloor \frac{MBCount_x}{2} \right\rfloor , \; y_{partition} = \left\lceil \frac{MBCount_y}{2} \right\rceil$$

where,

$MBCount_x$=total number of macroblocks along the x-axis

$MBCount_y$=total number of macroblocks along the y-axis
$x_{partition}$= vertical partition line after xth macroblock along the x-axis
$y_{partition}$= horizontal partition line after yth macroblock along the y-axis

The arrows in Fig.7.2b depict the macroblock count in the TL regions of the three frames being compared to generate the first row (vector) of the ordinal rank matrix. Similarly, the macroblock count from the remaining three regions can be used to realize the final ordinal matrix of Fig.7.2c.



(a) Frame division into 2×2 regions

8	15
15	15

17	15
14	15

14	6
8	10

(b) Count of relevant macroblocks within each region

$$\begin{array}{r} \text{TL} = \\ \text{TR} = \\ \text{BL} = \\ \text{BR} = \end{array} \begin{pmatrix} 3 & 1 & 2 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$$

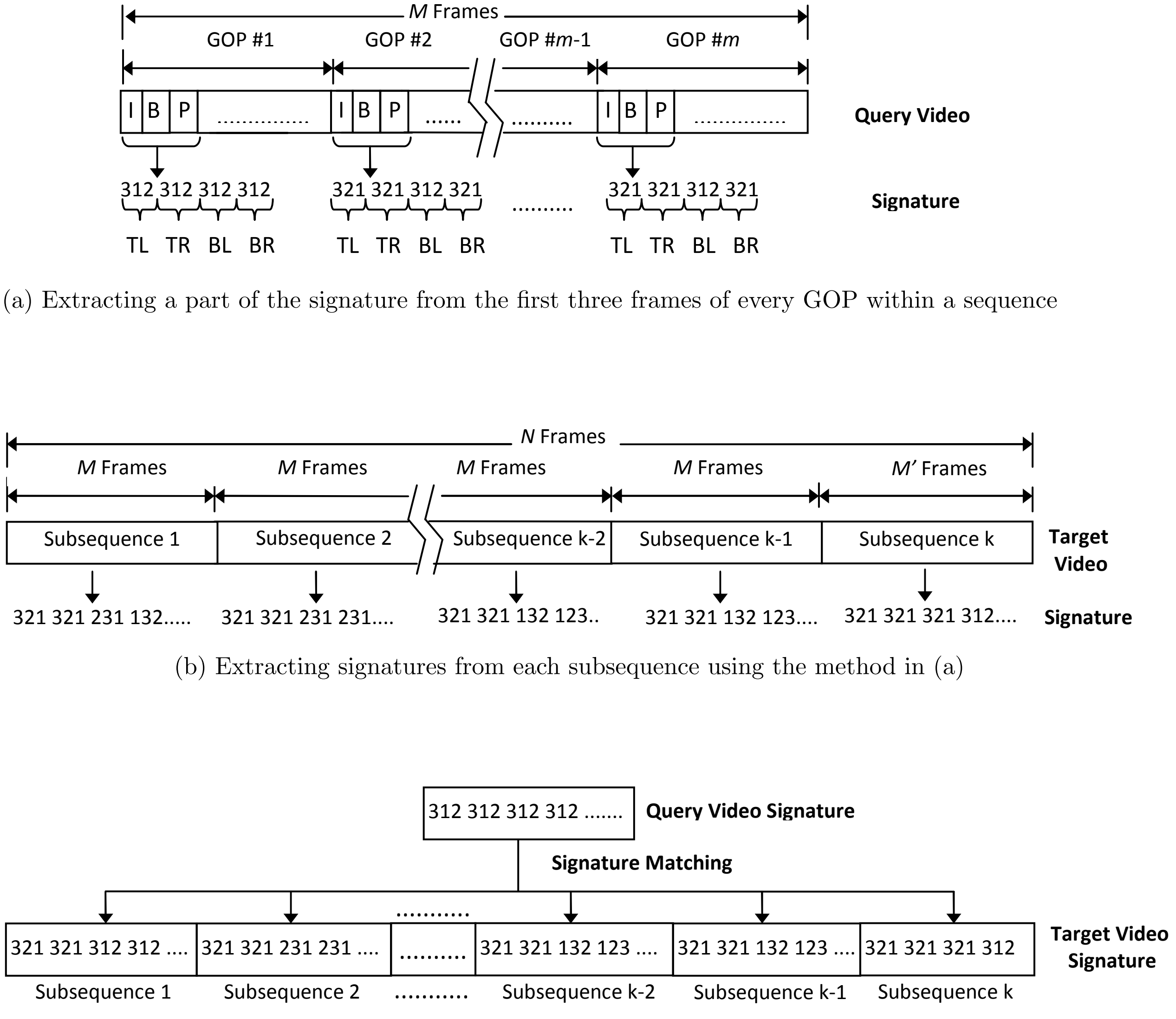
(c) Spatiotemporal ordinal matrix

Figure 7.2: Signature generation for an H.264 encoded video

7.3.4 Matching Methodology

Usually H.264/AVC coded video uses a GOP length of 12 or 24 frames [105] within the PAL standard. This GOP length provides an optimum balance between compression ratio and the quality of the video. Accordingly, for the experiments carried out, the query and the target video samples were divided into GOP lengths of 12. The first three frames at the beginning of every GOP were utilized to build a part of the signature. The justification behind choosing only three frames to build part-signatures was the simple fact that every H.264 encoded video sequence consists of only three types of frame: I,B and P. Using one frame of each type to build the part-signature was enough to capture the characteristics of a coded video sequence.

These part signatures were concatenated to constitute the complete signature of a given video sequence. The matching process was performed according to the mechanism shown in Fig.7.3. The signature is extracted from the query video clip as shown in Fig.7.3a. Similarly, signatures are extracted from every subsequence within the target video as shown in Fig.7.3b.It should be noted that each subsequence has the same GOP length as the target video except possibly the last (i.e. M′). Figure 7.3c depicts the matching process.



(a) Extracting a part of the signature from the first three frames of every GOP within a sequence

(b) Extracting signatures from each subsequence using the method in (a)

(c) Matching query video to target video for detecting copies

Figure 7.3: Signature generation and matching within H.264 video sequence

The problem of detecting copies of a given query video involves detecting not only whether a copy exists within a target video but also determining its location. We can formalize the problem by defining a few notations and symbols as follows. Let the query video be denoted as $V_Q = \{v_q^1, v_q^2, v_q^3, ..., v_q^m\}$ and the target video as $V_T = \{v_t^1, v_t^2, v_t^3, ..., v_t^n\}$ with M and N being the total number of frames within the query and target videos respectively and where, $M << N$. A subsequence within V_T is defined as V_T^r where $r \in [r : r + M - 1]$, with the number of frames being M and $0 \leq r \leq n - N$. Further let each frame within V_T or V_Q be denoted as $V_i = \{v_i[0], v_i[1], ...v_i[p]\}$ with P being the total number of partitions within each frame. Then the ranking matrix of partition p within V_Q $=\{v_{q,p}^1, v_{q,p}^2, v_{q,p}^3, ..., v_{q,p}^M\}$ can be denoted as $\pi_{q,p}$. The size of each $\pi_{q,p}$ would be of the order of $[1{\times}M]$. Similarly, any subsequence within the target video $V_T^r = \{v_{t,p}^r, v_{t,p}^{r+1}, v_{t,p}^{r+2}, ..., v_{t,p}^{r+M-1}\}$ can be denoted as $\pi_{t,p}^r$ also with a size of $[1{\times}M]$.

We can then define the problem of detecting copies as follows. Given a target video V_T, we say that a subsequence, V_T^r from V_T is a copy of V_Q if the distance $D(V_Q, V_T^r)$ is below a threshold $\epsilon \in [0, 1]$. The distance measure D is calculated as:

$$D(V_Q, V_T^r) = \frac{1}{P} \sum_{p=1}^{P} d(\pi_{q,p}, \pi_{t,p}^r)$$

and where d is calculated as:

$$d(\pi_{q,p}, \pi_{t,p}^r) = \frac{1}{C} \sum_{i=1}^{M} |\pi_{q,p}(i) - \pi_{t,p}^r(r + i - 1)|$$

Each d is the normalized distance between two rank matrices. C is the maximum distance between two rank matrices π_k and π_j ,$\forall(\pi_k, \pi_j) \in S_P$,with S_P being the set of all possible rank matrices with size P. C is obtained when the two rank permutations are reverse of each other. It is calculated as:

$$C = \sum_{i=1}^{M} |M + 1 - (2 \times i)|$$

For the experiments reported in this Chapter, we used $P = 4$; hence $C = 8$.The copy detection mechanism proceeds as shown in Algorithm 2.

A sample functioning of Algorithm 2 is shown in Fig.7.4. In this example, it is assumed that subsequence V_T^3 is a copy of V_Q. The matching and detection procedure deduces $D(V_Q, V_T^3)$ as being less than ϵ and accordingly flags the subsequence as a copy and pinpoints its location within V_T.

```
input  : $V_Q$=Query video clip;
         $V_T$=Target video sequence;
         $\epsilon$=Matching detection threshold;
output: Copied video sequence $V_T^{r*}$ and its location $i$ within $V_T$
Initialize $r = 0, i = 0, gop\_length = 12$, $D_v[n - N]$=0;
while $r \leq n$-$N$ do
    Calculate $D(V_Q, V_T^r)$ and store in $D_v[i]$;
    increase i by 1;
    increase r by $gop\_length$;
Locate minimum value $D_{min}$ within $D_v[i]$;
if $D_{min} < \epsilon$ then
    Declare corresponding $V_T^r$ as a copy and $i$ as the location of the copy
    within $V_T$;
```

Algorithm 2: Copy detection methodology

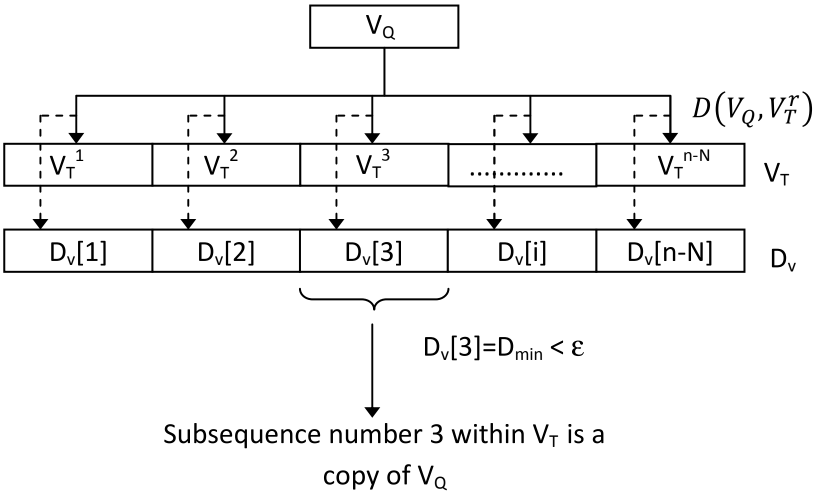


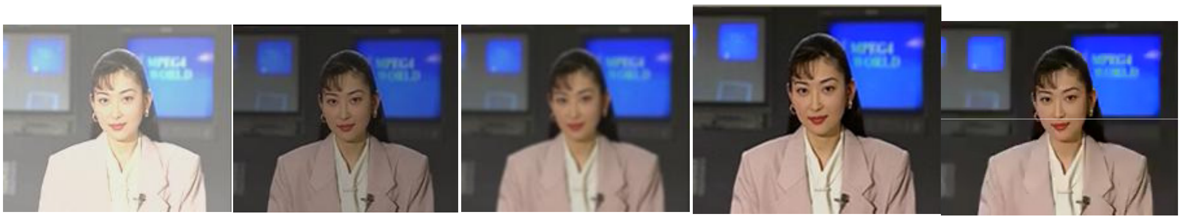
Figure 7.4: An instance of Algorithm 2

7.3.5 Compressed Domain Matching

The H.264 encoder generates an encoded bitstream in the form of slices. Each slice in turn contains a slice header and a slice payload. Each slice payload contains within it a number of macroblocks. Each macroblock has a header and a payload. The macroblock header contains the information regarding the macroblock type. Since signature generation within the proposed method is based on the number of macroblocks of a specific type; to match a video for detecting copies, it is only necessary to partially decode the H.264 bitstream, read the macroblock type within each of the 2×2 regions and construct the signature. Since the complexity of the H.264 codec is quite high, complete decoding of H.264 video bitstream, extracting the signature and then re-encoding it would incur a significant computational cost. Designing a compressed domain CBCD system avoids this overhead. Finally, compressed domain CBCD systems are becoming more popular for all video standards since nearly all of the video content at present is in compressed form, either for transmission, distribution or storage. However, H.264 has been largely ignored as a standard when it comes to CBCD systems. This work is an attempt bridge a part of this gap.

7.4 Experimental Results

The system was implemented within the JM Reference Software version 15.1 [62]. A requirement of using this software was that it accepts only raw YUV files as input. It was not possible to obtain a significant database of YUV files to test the proposed system as there are only limited numbers of YUV QCIF (176×144) video sequences available online [106]. A set of 24 different video sequences with different lengths were however obtained. These video sequences covered almost all subjects like news, sports, scenery, architecture, interviews etc. These were concatenated to realize a longer target video sequence of 13,372 frames. Subsequences of length 30, 50, 100 and 150 frames were randomly selected from the above video sequence and 13 different transformations were applied to simulate copied video sequences. They were: increase brightness by 25%, decrease brightness by 25%, increase contrast by 25%, decrease contrast by 25%, decrease frame size down to 80%, increase frame size up to 120%, temporal smoothening, motion blurring, Gaussian radius-2 blurring, general convolution, decreasing frame rate down to 0.8 times the original rate, increasing frame rate upto 1.2 times the original rate and letter box. A few snapshots of these transformations are shown in Fig.7.5. Query videos clips of lengths 30, 50, 100 and 150 frames were then matched to the target video using the procedure outlined in the previous section.



(a) Brighter by 25%

(b) Contrast reduced by 25%

(c) Gaussian radius-2 blur

(d) Frame resized to 210×174

(e) Original

Figure 7.5: Some of the transformations applied to generate query video sequences and the original video

The proposed algorithm was tested for varying values of ϵ and the receiver operating characteristics (ROC) curve was plotted as shown in Fig.7.6. This plot depicts the false positive rate (FPR) versus the false negative rate (FNR). These rates are calculated as follows. Let F_N be the number of false negatives i.e. number of copy-clips undetected and F_P be the number of false positives i.e. number of non-copy clips detected as a copy. Further, let N_T be the total number of non-copy clips and N_C be the total number of copy clips. Then for a specific value of ϵ, the FPR and FNR can be calculated as:

$$\text{FNR}(\epsilon) = \frac{F_N}{N_C} \text{ , } \text{FPR}(\epsilon) = \frac{F_P}{N_T}$$

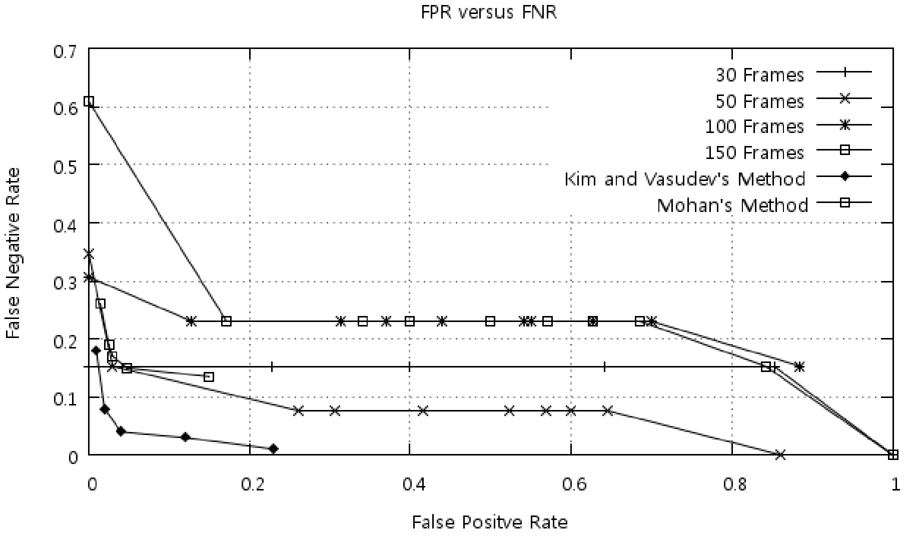


Figure 7.6: ROC curve for FPR versus FNR

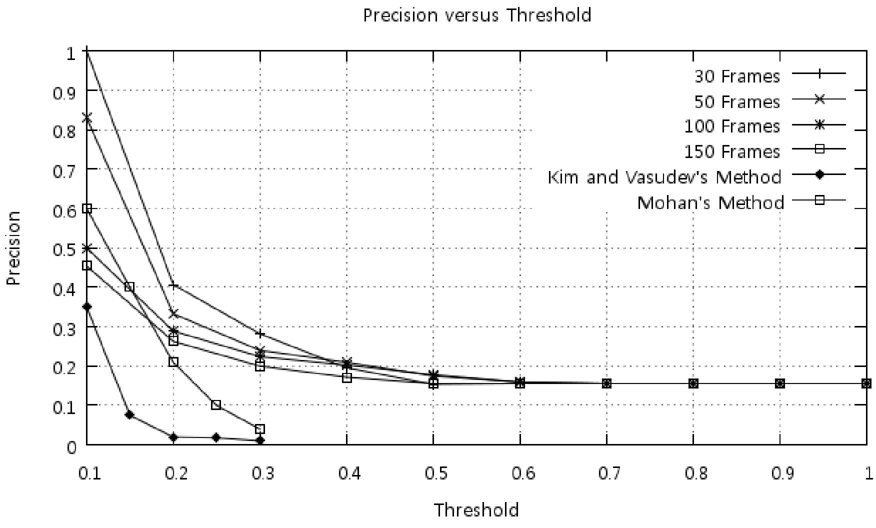
The ideal ROC curve would pass through the origin. This implies that closer the ROC curve passes by the origin; the better is the performance of the algorithm. As can be seen from Fig.7.6, the ROC curve for the proposed technique is closest to the origin when the video length is 50 frames. Interestingly, when the query video length is either decreased to 30 or increased to 100 frames and then further up to 150 frames, the ROC curve moves further away from the origin. Particularly, using a 30 frame query video length fails to capture the spatiotemporal similarities between an original video clip and its copy thereby leading to a higher false negative rate. As a result, the query video length for the proposed CBCD system can be fixed at 50 frames to guarantee optimum performance. Figure 7.6 also indicates that the FPR and FNR rates for the proposed algorithm are higher than the technique proposed by Kim and Vasudev [49] but lower than the one proposed by Mohan [107]. However, the proposed algorithm is computationally much more efficient than both the above mentioned approaches. The above claim is due to two factors. First, the matching process within the proposed technique uses only 3 frames per GOP in contrast to every frame as was reported in [49] and [107]. Second, the proposed technique is a compressed domain algorithm while the other two are spatial domain techniques.

Another measure of the performance of any CBCD algorithm is to compute the precision and the recall rates for varying threshold values. These parameters are calculated as follows:

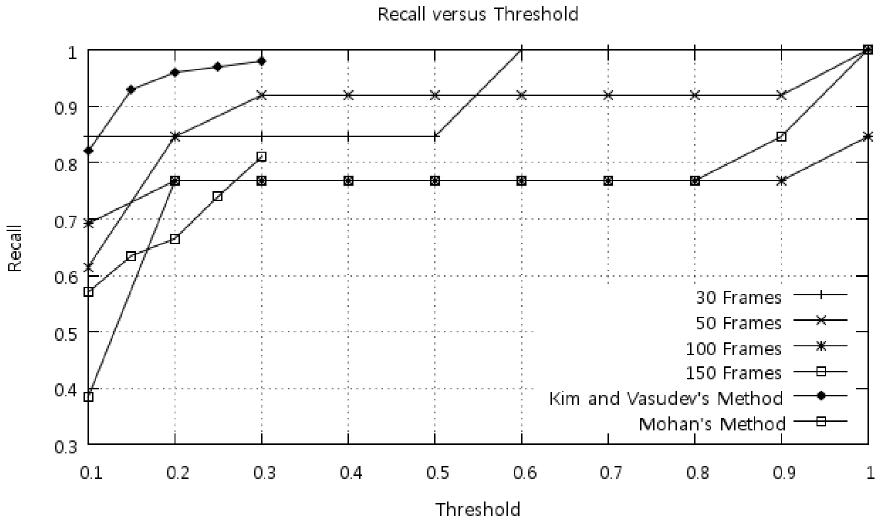
$$\text{Precision}(\epsilon) = \frac{\text{number of copy clips successfully detected leaving out false positives}}{\text{number of clips detected as a copy including false positives}}$$

$$\text{Recall}(\epsilon) = \frac{\text{number of copy clips successfully detected leaving out false positives}}{\text{total number of actual copy clips}}$$

Fig.7.7 shows the precision and recall rates plotted against normalized threshold values. The plot also includes the precision and recall rates reported in [49] and [107]. Kim and Vasudev [49] compared their method to the one proposed in [107] at a threshold value of ϵ=0.1. It can be seen from Fig.7.7 that at this threshold value, both the precision and recall rates for the proposed technique are higher than those reported in [49] and [107]. Even though the precision and recall rates are highest at a video query length of 30 frames, due to the poor ROC obtained (refer to Fig.7.6), we claim that optimum performance from the proposed system can be obtained at a query length of 50 frames when $\epsilon = 0.1$.



(a)



(b)

Figure 7.7: Precision and recall rates versus normalized threshold

Fig.7.8 shows the precision and recall rates for the proposed technique at different query lengths as well the rates reported in [49]and [107]. These values have again been obtained at ϵ= 0.1. As can be seen, the proposed method offers very encouraging results. Even though the recall rate is comparable to the method proposed in [49] however, the precision rate is much higher which signifies that there are less false detections within the proposed system even with such short length query videos.

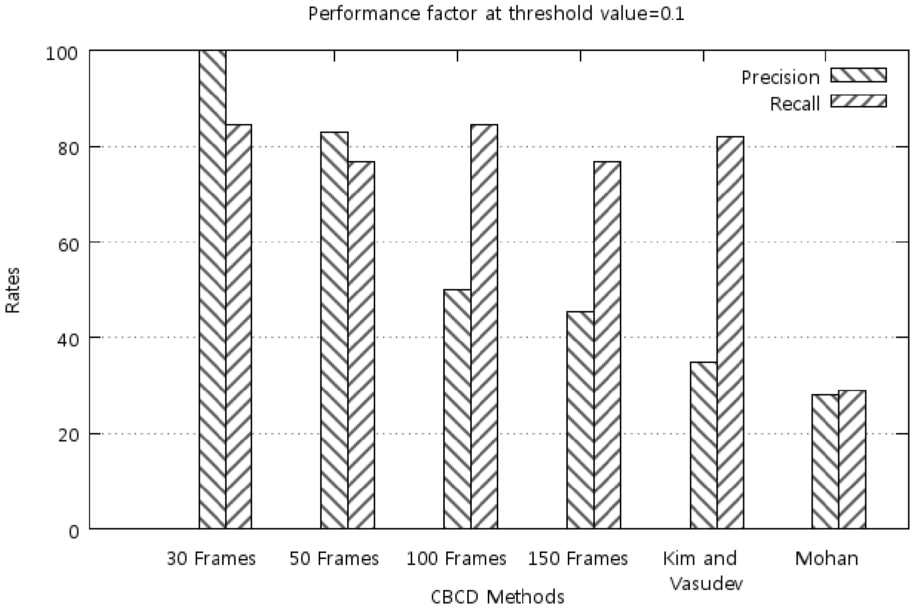


Figure 7.8: Performance evaluation by measuring precision and recall rates

The final evaluation of the proposed technique was in terms of the time taken to detect a copy. It is obvious that the proposed technique would be efficient since only the first 3 frames of each GOP within a sequence are used to generate the signature in contrast to every frame. In particular only 3343 frames out of the above mentioned video sequence of 13,372 frames will play a role in the matching process. The copy detection time is computed from the time the signature is extracted out of the frames and matched. As per this condition, the time taken to match a 50 frame query video to the target video on a Pentium 4 PC, 3.4 GHz and 3GB of RAM was 0.38 seconds. The low detection time is also due to the fact that only partial decoding of the H.264/AVC bitstream is required in order to count the macroblock type.

Finally, looking at the memory requirements, generating a three frame part-signature within each GOP and with each frame having 4 partitions gives a size of 12 bytes. With a GOP size of 12, a 50 frame query video sequence would have 5 GOP boundaries. This would give a total signature size of 60 bytes per video clip, which is significantly low when considering memory and storage requirements.

7.5 Conclusion

This chapter presented a compressed domain CBCD system designed to detect video copies encoded with the H.264/AVC standard. The proposed technique utilized a feature unique to the H.264/AVC standard wherein different regions of a frame and different types of frames are encoded as different macroblock types. The proposed technique uses a short sequence of 3 frames (I, B, P) at the beginning of every GOP within a given video clip to construct a spatiotemporal signature. This is done by reading the various macroblock types from a partially decoded H.264 bitstream. This method of signature extraction makes the proposed technique not only computationally efficient but also resistant to common video editing effects such as frame rate change/frame dropping. In addition, use of ordinal methods to construct the signature guarantees that the proposed technique is resistant towards frame resizing, letter-box and other common video processing steps. Finally, since the proposed technique is a compressed domain method with a low processor and memory footprint, it is also suitable for devices with limited computing resources such as smartphones, PDAs or portable video players. Future work could include a look into using other macroblock types such as 16×8 or 8×16 in order to develop a more accurate and robust signature. The result of this work have been submitted for publication at [108] and is currently under review.

Chapter 8

Conclusion

This work was an attempt to address the issue of Digital Rights Management (DRM) specifically for H.264 Video. DRM can be applied using a number of techniques and some of these were explored within the scope of this work. Section 8.1 highlights the milestones and deliverables achieved during the course of this work while section 8.2 provides suggestions for future work.

8.1 Milestones and Deliverables

Using computationally intensive techniques such as encryption to enforce DRM encounters hurdles when it comes to H.264 video. This is due to the fact that the standard caters to a large class of application domains and devices some of which may not possess sufficient computational resources to decrypt the content in a reasonable amount of time and commence playback. Recognizing this aspect, it was decided during the course of this work, to explore other computationally efficient options to enforce DRM. The resulting work proposed a number of methods based on content and copyright protection, content authentication and content based copy detection.

The robust watermarking algorithm proposed in Chapter 4 was designed to protect H.264 content and to verify proof of ownership. This was done by utilizing the DC residuals within 4×4 intrapredicted macroblocks. It was proposed that since DC residuals are resistant towards most of the common signal processing steps, a watermark embedded within them would be robust enough to repel any attacks. Further, such a watermark would also be resistant towards compression. The proposed technique was evaluated for its R-D characteristics which were found to be encouraging while the watermarked video suffered no significant quality degradation. The technique was proven to be resistant to common signal processing attacks. However, the technique suffers from two drawbacks: (1)

Since the choice of DC residuals to carry the payload is made randomly, a "key" which identifies these macroblocks (containing the chosen DC residual) has to be sent across to the decoder in order to extract the watermark. This constitutes supplementary information and hence the technique is not self-contained. (2) The technique would be vulnerable against an attack such as transcoding which could change the prediction mode and hence the residual values. This would mean failure to extract the watermark. However, this technique offers better robustness than the technique proposed in [58] wherein AC residuals are used as embedding locations.

In Chapter 5 and 6, an attempt was made to explore another aspect of watermarking i.e. fragile watermarks. This category of watermark is used for content authentication. Fragile watermarks can be irreversible and reversible. The technique proposed in Chapter 5 was an irreversible watermarking approach while the one proposed in Chapter 6 was a reversible one. The method presented in Chapter 5 operates during the CAVLC entropy encoding stage by embedding the watermark bits within the last coefficient of a 4×4 quantized block of an I-frame. Since the embedding takes place after entropy encoding and just before bitstream formation, it is essentially a compressed domain approach. In addition, due to the embedding being performed within the highest AC coefficients, any unauthorized modifications to the host video would easily perturb the watermark and hence indicate tampering. Thus the effectiveness of this method lies in not just its ability to authenticate a genuine user but also to detect tampering, if any. Attacks were simulated on the proposed method and the watermark extracted to check for tampering. A detector was set up at the receiver side and it was found that the technique is sensitive enough to detect most of the attacks such as transcoding, rotation, median filtering and cropping. The technique was also compared with the techniques in [69, 72–74] and was found to exhibit better R-D characteristics for a wide range of quantization parameters. The use of simple mathematical operations also guarantees its computation efficiency.

Chapter 6 discussed a reversible watermarking technique based on Difference Expansion (DE). Since all reversible watermarking techniques are fragile in nature, the method proposed in this chapter is also useful for content authentication. In this method, the technique of DE was applied on IPCM macroblocks within P-frames to embed a high capacity payload. Using P-frames only served two purposes: (1) Reduces the amount of drift since they are not used as frequently as I-frames for interprediction. (2) P-frames occur more frequently than I-frames and hence offer more embedding "space". The technique of DE offers multi-layer watermark embedding which is unique. The proposed technique utilized both, the luminance and the chrominance components to embed the payload in a reversible way. The performance of the proposed technique was evaluated under a number of encoding parameters, namely, with the rate control OFF and ON. The variation in PSNR and bitrate under increasing payloads was observed and it was found that the performance of the proposed system remain stable and does not degrade significantly. The computational overhead involved as a result of embedding the watermark was also evaluated and the observations showed that the total increase was not more than 3% over the time taken by a "normal" H.264 encoder. Finally, the proposed technique was compared with other similar techniques in terms of R-D characteristics [88] and payload capacity [88,90].It was found that the proposed technique exhibited better R-D characteristics while at the same time allowing a payload capacity that was almost twice of the other two reported techniques. Such a technique can be put to use in a number of applications such as Video-on-Demand where the source could be authenticated before accepting a video stream. After authentication and before commencing playback, the video could be restored back to its original form.

In Chapter 7, the problem of DRM was looked at from a completely different perspective. Rather than embedding additional information within the video and then extracting it at the receiver side, the characteristics of the video itself were utilized to generate a signature that would uniquely identify the video. In this approach, a group of I-, B- and P-frames at the start of each GOP were divided into 2×2 regions. Then the number of 4×4, B_skip and P_skip macroblocks were counted within each region of each frame respectively. The values were then sorted in an ordinal matrix in order to constitute a signature. Signatures were extracted from a query video clip and matched against signatures extracted from longer target video sequences. A distance parameter was calculated that measured the similarity between two signatures. A distance value lesser than a pre-defined threshold would indicate a match, and hence a copy. This method was tested on a number of test video sequences encoded under the H.264 standard. "Copies" of video sequences were generated by performing some of the most common video editing operations. They included not only spatial but also temporal modifications such as changes in luminance, contrast, frame rate change and motion blurring etc. In total, 13 different types of video editing steps were performed in order to generate copies. It was found that the method is capable of detecting,on an average, 10 out of 13 copies even with a low detection sensitivity. In addition, the technique was also capable of temporal localization which implies that the technique was able to identify the location within a longer video sequence where the copy existed. Analytically, the performance was evaluated by plotting an ROC curve for false-positive and false-negative rates. The results indicated that the technique provided optimum performance when the sequence matching length was

50 frames. A query length lower or higher than this exhibited less than optimum ROC characteristics. Another set of parameters used to evaluate CBCD based systems was *precision* and *recall*. These factors were plotted against normalized threshold values and again the technique reported satisfactory results. In fact, when compared to two other similar methods reported in [49, 107], the proposed technique exhibited a much higher precision rate while maintaining a comparable recall rate. This proved that there were far less false detections. Even though the precision and recall rates with a frame length of 30 were quite satisfactory but in the light of the poor ROC curve obtained, the optimum performance of the proposed technique was settled at a query length of 50 frames. Finally, the computational efficiency of the technique was measured in terms of the time taken to detect a copy. It was found that technique is very efficient and takes less than half a second to identify a 50 frame query video in a target video sequence of 13,372 frames.

8.2 Suggestions and Future Work

There are a number of avenues that could be further explored within the techniques proposed in this work. They range from improving only certain specific aspects of an algorithm to a complete re-design. Some of the obvious avenues for further exploration are:

1. The robust watermarking technique proposed in Chapter 4 can be redesigned to make it self-contained. This essentially means that the key (which is supplementary information) can be made a part of the payload. Another possibility is to make the technique location-unaware. The framework to design a location unaware watermark detection have already been proposed in [109, 110]. A similar framework can be employed so that the watermark detector need not know which DC residuals are actually carrying the payload in order to extract the watermark
2. The fragile watermarking approach presented in Chapter 5, can be improved by generalizing the choice of watermark embedding locations. Thus every residual block can contain a watermark bit in not just the last coefficient but in any of the last 3-4 coefficients. This will increase not just the security of the algorithm but also make it resistant towards those categories of attacks that attempt to completely obliterate the watermark.
3. The reversible watermarking approach presented in Chapter 6 can be considered to be only a basic design. There are several ways the technique can be improved upon:
   (a) Introducing a heuristic method that makes a more informed decision

regarding which macroblock to encode as an IPCM macroblock. This decision can be taken on the basis of the texture/detail/motion information contained within the macroblocks.

(b) The pixels can be paired using any pattern other than a simple horizontal or vertical pattern. This will improve the security of the algorithm.

(c) Using B-frames in addition to P-frames to embed the payload. This will naturally improve the payload capacity.

(d) Introducing a drift compensation module.

4. The copy detection method proposed in Chapter 7 can be improved by including other macroblocks types that are generated by the H.264 encoder. They include the 8×8 macroblocks generated within the I-frame; 8×16, 16×8 macroblocks generated within the B- and P-frames etc. Using different macroblocks types will provide a more accurate signature which in turn can lead to a better performance. Further, a combination of the count and intensity/DCT values within these macroblocks can also be used to generate a spatiotemporal signature. However, whether it will improve copy-detection performance still remains an open research problem.

5. The methods developed within this study can be combined together to realize a more effective DRM system. For instance, the robust and fragile watermarking methods can be combined to realize a hybrid watermarking system. Such a system will be effective for both content protection and content authentication. Similarly, a CBCD system can be combined with a robust watermarking system. Such a system would first, be able to detect a copy of the original video and then extract the robust watermark from the copy in order to prove ownership.

Enforcing DRM especially on H.264 video is rapidly gaining attention both within the academia and industry. This is mainly due to the rapid growth of high-speed broadband and cellular networks which allows more and more users to access and consume video content. This also means that unauthorized users and pirates who have access to powerful tools and resources can easily copy, modify and redistribute the video. It seems that the pirates are always a step ahead of the latest DRM technology, thus newer and powerful techniques are always required in order to protect digital video.

Bibliography

[1] An analysis prepared by LEK for the Motion Picture Association, "The cost of movie piracy." Available at: http://austg.com/include/downloads/PirateProfile.pdf, 2005.

[2] Joint Video Team (JVT) of MPEG and VCEG, "Draft ITU-T recommendation and final draft international standard joint video specification," *From: Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6)*, Mar 2003.

[3] G. Sullivan, P.N.Topiwala, and A.Luthra, "The H.264/AVC advanced video coding standard: overview and introduction to the fidelity range extensions," in *SPIE*, vol. 5558, pp. 454–476, 2004.

[4] T.Wiegand, G. J. Sullivan, J. Reichel, H. Schwarz, and M.Wien, "Joint draft 11 of SVC amendment," Doc. JVT-X201, Joint Video Team, July 2007.

[5] I.E.G Richardson, *H.264 and MPEG-4 Video Compression.* Wiley,Chichester, 2004.

[6] A.M.Tourapis, F.Wu, and S.Li, "Direct mode coding for bipredictive slices in the H.264 standard," *IEEE Transactions on Circuits and System for Video Technology*, vol. 15, pp. 119–126, Jan. 2005.

[7] ITU-T Recommendation H.264 ISO/IEC 14496-10 AVC, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification," *From: Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6)*, vol. 8th Meeting: Geneva, Switzerland, 23-27 May 2003.

[8] S. Wenger, "H.264 over IP," *IEEE Transactions on Circuits and System for Video Technology*, vol. 13, no. 7, pp. 645–656, 2003.

[9] T.Stockhammer, M. Hannuksela, and T.Wiegand, "H.264 in wireless environments," *IEEE Transactions on Circuits and System for Video Technology*, vol. 13, no. 7, pp. 657–673, 2003.

[10] Q.Liu, R.Safavi-Naini, and N.P.Sheppard, "Digital rights management for content distribution," in *Australasian information security workshop conference on ACSW frontiers* (C. Johnson, P. Montague, and C. Steketee, eds.), vol. 21, (Australia), pp. 49–58, Darlingtonhurst, 2003.

[11] E.I.Lin, A.Eskicioglu, R.Lagendijk, and E.Delp, "Advances in digital video content protection," in *Proceeding of IEEE*, vol. 93, pp. 171–183, Jan. 2005.

[12] G. Hobbs, "Video scrambling." U.S. Patent 5 815 572, Sept.29 1998.

[13] D. Zeidler and J. Griffin, "Method and apparatus for television signal scrambling using block shuffling." U.S. Patent 5 684 876, June14 1994.

[14] H.Pinder and M.Palgon, "Apparatus and method for cipher stealing when encrypting MPEG transport packets." U.S. Patent 5 684 876, Nov. 4, 1997.

[15] I.Agi and L.Gong, "An empirical study of secure MPEG video transmissions," in *The Internet Society Symposium on Network and Distributed System Security*, Feb. 1996.

[16] T.Maples and G.Spanos, "Performance study of a selective encryption scheme for the security of networked, real-time video," in *4th International Conference on Computer Communications and Networks*, (Las Vegas,Nevada,USA), Sept. 1995.

[17] J.Meyer and F.Gadegast, "Security mechanisms for multimedia data with the example of MPEG-1 video." Available at: http://www.cs.tuberlin.de/phade/phade/secmpeg.html, 1995.

[18] Y.Zou, T.Huang, W.Gao, and L.Huo, "H.264 video encryption scheme adaptive to DRM," *IEEE Transactions on Consumer Electronics*, vol. 52, p. 1289, 1297 Nov.2006.

[19] S.-W. Park and S.-U. Shin, "Efficient selective encryption scheme for the H.264/Scalable Video Coding(SVC)," in *Fourth International Conference on Networked Computing and Advanced Information Management NCM'08*, pp. 371–376, Sept. 2008.

[20] R.Iqbal, S.Shirmohammadi, and A.E.Saddik, "Compressed-domain encryption of adapted H.264 video," in *Eighth IEEE International Symposium on Multimedia ISM'06*, pp. 979–984, Dec. 2006.

[21] D.Boneh, "Cryptographic hashing."
http://crypto.stanford.edu/$\tilde{d}$abo/courses/cs255_winter01/1-hashing.pdf
[Online: 09 May 2011].

[22] B.Coskun, B.Sankur, and N.Memon, "Spatio–temporal transform based video hashing," *IEEE Transactions on Multimedia*, vol. 8, pp. 1190–1208, Dec. 2006.

[23] J.Oostveen, T.Kalker, and J.Haitsma, "Visual hashing of digital video: applications and techniques," in *SPIE Applications of Digital Image Processing XXIV*, (San Diego,CA,USA), July 2001.

[24] N.Ramaswamy and K. R. Rao, "Video authentication for H.264/AVC using digital signature standard and secure hash algorithm," in *2006 international workshop on Network and operating systems support for digital audio and video (NOSSDAV'06), ACM*, vol. 21, (NY,USA), pp. 1–6, 2006.

[25] X.Wang, N. Zheng, and L.Tian, "Hash key-based video encryption scheme for H.264/AVC," *Signal Processing: Image Communication*, vol. 25, pp. 427–437, Jul. 2010.

[26] I.J.Cox and M.L.Miller, "Electronic watermarking: the first 50 years," in *IEEE Fourth Workshop on Multimedia Signal Processing*, pp. 225–230, 2001.

[27] V.Schyndel, R. Tirkel, and A.Z.Osborne, "A digital watermark," in *IEEE International Conference on Image Processing (ICIP '94)*, vol. 2, (Austin,Texas,USA), pp. 86–90, Nov.13-16 1994.

[28] N.Nikoliadis and I.Pitas, "Copyright protection of images using robust digital signatures," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'96)*, vol. 4, (Lausanne, Switzerland), pp. 2168–2171, Sept. 1996.

[29] I.Pitas, "A method for signature casting on digital images," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'96)*, vol. 3, pp. 215–218, Sept. 1996.

[30] P.Wolfgang and E. Delp, "A watermark for digital images," in *IEEE International Conference on Image Processing (ICIP'96)*, vol. 3, pp. 219–222, 16-19 Sept. 1996.

[31] F. Boland, J. Ruanaidh, and C. Dautzenberg, "Watermarking digital images for copyright protection," in *IEE Conference on Image Processing and Applications*, pp. 326–331, July 1995.

[32] G.Bors and I.Pitas, "Image watermarking using DCT domain constraints," in *IEEE International Conference on Image Processing(ICIP'06)*, vol. 3, (Lausanne, Switzerland), pp. 231–234, 16-19 Sept. 1996.

[33] C.T.Hsu and J.L.Wu, "Hidden signature in images," in *IEEE International Conference on Image Processing (ICIP'96)*, vol. 3, (Lausanne, Switzerland), pp. 223–226, 16-19 Sept. 1996.

[34] B.G. Mobasseri and R.J.Berger II, "A foundation for watermarking in the compressed domain," *IEEE Signal Processing Letters*, vol. 12, pp. 399–402, May 2005.

[35] F.Hartung and B.Girod, "Watermarking of uncompressed and compressed video," *Signal Processing*, vol. 66, pp. 283–301, May 1998.

[36] R. Chandramouli, N. D. Memon, and M. Rabbani, *Digital Watermarking.* New York: Wiley, 2002.

[37] M. L. Miller, G. J. Doerr, and I. J. Cox, "Applying informed coding and embedding to design a robust, high capacity watermark," *IEEE Transaction on Image Processing*, vol. 13, pp. 792–807, June 2004.

[38] J.Eggers and B.Girod, *Informed Watermarking*, vol. 685. 1402070713: Kulwer Academic, 2002.

[39] J.Eggers and B.Girod, "Blind watermarking applied to image authentication," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'01)*, vol. 3, pp. 1977–1980, 2001.

[40] L.Tian, N.Zheng, J.Xue, and T.Xu, "A CAVLC-Based Blind Watermarking Method for H.264/AVC Compressed Video," in *IEEE Asia-Pacific Conference on Services Computing(APSCC'08)*, pp. 1295–1299, 2008.

[41] F. Mintzer, G. Braudaway, and M.Yeung, "Effective and ineffective digital watermarks," in *IEEE International Conference on Image Processing (ICIP'97)*, pp. 9–12, Oct. 1997.

[42] F.Mintzer, G.Braudaway, and A.Bell, "Opportunities for watermarking standards," *Communications of the ACM*, vol. 41, pp. 57–64, July 1998.

[43] G.Friedman, "The trustworthy digital camera: Restoring credibility to the photographic image," *IEEE Transactions on Consumer Electronics*, vol. 39, pp. 905–910, Nov. 1993.

[44] I.F.Kallel, M.Kallel, and M. Bouhlel, "A secure fragile watermarking algorithm for medical image authentication in the DCT domain," in *Information and Communication Technologies (ICTTA'06)*, pp. 2024–2029, 2006.

[45] C.Kim, "Ordinal measure of DCT coefficients for image correspondence and its application to copy detection," in *SPIE- IS&T Storage and Retrieval for Media Databases*, (Santa Clara,USA), pp. 199–210, Jan. 2003.

[46] E.Y.Chang, J. Wang, C. Li, and G. Wiederhold, "RIME: A replicated image detector for the World-Wide-Web," in *SPIE Symposium of Voice, Video, and Data Communications*, pp. 58–67, Nov 1998.

[47] M.Naphade, M.Yeung, and B.Yeo, "A novel scheme for fast and efficient video sequence matching using compact signatures," in *SPIE Conference on Storage and Retrieval for Media Databases*, vol. 3972, pp. 564–572, Jan. 2000.

[48] D.N.Bhat and S. Nayar, "Ordinal measures for image correspondence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 415–423, Apr. 1998.

[49] C.Kim and B.Vasudev, "Spatiotemporal sequence matching for efficient video copy detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, pp. 127–132, Jan. 2005.

[50] L.Chen and F. W. M. Stentiford, "Video sequence matching based on temporal ordinal measurement," *Pattern Recognition Letters*, vol. 19, pp. 1824–1831, Oct. 2008.

[51] R.Nie, G. Ding, J.Wang, and L.Zhang, "A new fingerprint sequences matching algorithm for content-based copy detection," in *Fifth International Conference on Information Assurance and Security (IAS'09)*, vol. 1, pp. 427–430, 18-20 Aug. 2009.

[52] A.Basharat, Y.Zhai, and M.Shah, "Content based video matching using spatiotemporal volumes," *Computer Vision and Image Understanding*, vol. 110, pp. 360–377, June 2008.

[53] A.Hampapur and R.M.Bolle, "Comparison of distance measures for video copy detection," in *IEEE International Conference on Multimedia and Expo ICME*, pp. 737–740, 22-25 Aug. 2001.

[54] Z.Li and J.Chen, "Efficient compressed domain video copy detection," in *International Conference on Management and Service Science MASS*, vol. 1-4, 24-26 Aug. 2010.

[55] E.Koch, J.Rindfrey, and J. Zhao, “Copyright protection for multimedia data,” in *International Conference on Digital Media and Electronic Publishing*, (Leeds,UK), 1994.

[56] J. R. Smith and B. O. Comiskey, “Modulation and information hiding in image,” in *Information Hiding International Workshop*, (Cambridge,UK), pp. 207–226, May 1996.

[57] M.D.Swanson, B.Zhu, and A.H.Tewfik, “Robust data hiding for images,” in *IEEE Digital Signal Processing Workshop*, (Loen,Norway), pp. 37–40, Sept. 1996.

[58] M. Noorkami and R. Mersereau, “Compressed domain video watermarking for H.264,” in *IEEE International Conference on Image Processing (ICIP‘05)*, vol. 2, (Singapore), pp. 890–893, Sept. 2005.

[59] G.-Z. Wu, Y.-J. Wang, and W.-H. Hsu, “Robust watermark embedding/detection algorithm for H.264 video,” in *Journal of Electronic Imaging*, vol. 14, p. 013013, Jan-Mar 2005 2005.

[60] P.Meerwald and A.Uhl, “Robust watermarking of H.264-encoded video: Extension to SVC,” in *Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, pp. 82–85, 15-17 Oct. 2010.

[61] X.Gong and H-M.Lu, “Towards fast and robust watermarking scheme for H.264 video,” in *Tenth IEEE International Symposium on Multimedia (ISM‘08)*, pp. 649–653, 15-17 Dec. 2008.

[62] Reference JVT:, “Software version 15.0.” http://iphome.hhi.de/suehring/tml/download/, Apr. 2009.

[63] M.A.Ali and E.A.Edirisinghe, “Watermarking H.264/AVC by modifying DC coefficients,” in *International Conference on Cyberworlds, (CW‘09)*, (Bradford,UK), pp. 241–245, 7-11, Sept. 2009.

[64] D. Kundur and D. Hatzinakos, “Towards a telltale watermarking technique for tamper-proofing,” in *IEEE International Conference on Image Processing (ICIP‘98)*, vol. 2, (Chicago,Illinois,USA), pp. 409–413, 4-7 Oct. 1998.

[65] M.Wu and B.Liu, “Watermarking for image authentication,” in *IEEE International Conference on Image Processing(ICIP‘98)*, vol. 2, (Chicago,Illinois,USA), pp. 437–441, 4-7 Oct. 1998.

[66] D. Stinson, *Cryptography Theory and Practice.* CRC Press,Boca Raton, 1995.

[67] E.T.Lin and E.J.Delp, "A review of fragile image watermarks," in *Multimedia and Security Workshop (ACM Multimedia'99)*, (Orlando,USA), pp. 25–29, Oct. 1999.

[68] T-Y.Chen, T-H.Chen, Y-T.Lin, Y-C.Chang, and D-J.Wang, "H.264 video authentication based on semi-fragile watermarking," in *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, (Harbin,China), pp. 659–662, Aug. 2008.

[69] T-Y.Kuo, Y-C.Lo, and C-I.Lin, "Fragile video watermarking technique by motion field embedding with rate-distortion minimization," in *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, vol. 853-856, (Harbin,China), Aug. 2008.

[70] C-C.Wang and Y-C.Hsu, "Fragile watermarking for H.264 video stream authentication," in *Eighth International Conference on Intelligent Systems Design and Applications (ISDA'08)*, vol. 1, pp. 77–80, 26-28 Nov. 2008.

[71] D.Pröfrock, H.Richter, M.Schlauweg, and E.Müller, "H.264/AVC video authentication using skipped macroblocks for an erasable watermark," in *SPIE Visual Communication and Image Processing*, vol. 5960, pp. 1480–1489, 2005.

[72] G.Qiu, P.Marziliano, A. Ho, D.He, and Q.Sun, "A hybrid watermarking scheme for H.264/AVC video," in *IEEE International Conference on Pattern Recognition*, vol. 4, pp. 865–868, 2004.

[73] Z.Zhu, G.Jiang, M.Yu, and X.Wu, "New algorithm for video watermarking," in *IEEE International Conference on Image Processing (ICIP'02)*, vol. 1, pp. 760–763, 2002.

[74] J.Zhang, J.Li, and L.Zhang, "Video watermark technique in motion vector," in *IEEE International Conference on Computer Graphics and Image Processing*, pp. 179–182, 2001.

[75] M.A.Ali and E.A.Edirisinghe, "A semi-fragile watermarking technique for H.264/AVC using CAVLC," *International Journal of Signal and Image Processing*, vol. Hypersciences, pp. 151–159, May 2010.

[76] J.Tian, "Reversible data embedding using differential expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 890–896, Aug. 2003.

[77] C.W.Hosinger, P.Jones, M.Rabbani, and J.C.Stoffel, "Lossless recovery of an original image containing embedded data," *US Patent, No 77102/E-D*, 1999.

[78] W.Bender, D.Gruhl, N.Morimoto, and A.Lu, "Techniques for data hiding," *IBM Systems Journal*, vol. 35, no. 3-4, pp. 313–336, 1996.

[79] J. Fridrich, J. Goljan, and R. Du, "Invertible authentication," in *SPIE, Security and Watermarking Multimedia Content*, vol. 1, (San Jose), pp. 197–208, Jan. 2001.

[80] C. D. Vleeschouwer, J.F.Delaigle, and B.Macq, "Circular interpretation of histogram for reversible watermarking," in *IEEE 4th Workshop on Multimedia Signal Processing*, (France), pp. 345–350, Oct. 2001.

[81] M.U.Celik, G.Sharma, A. Tekalp, and E.Saber, "Lossless generalized-LSB data embedding," *IEEE Transactions on Image Processing*, vol. 14, pp. 253–266, Feb. 2005.

[82] T.Kalker and F.M.J.Willems, "Capacity bounds and constructions for reversible data hiding," in *International Conference on Digital Signal Processing*, vol. 1, pp. 71–76, Jul. 2002.

[83] B. Macq, "Lossless multi-resolution transform for image authentication watermarking," in *EUSIPCO*, (Tampere,Finland), pp. 533–536, Sept. 2000.

[84] J. Tian, "Wavelet-based reversible watermarking for authentication," in *SPIE security and watermarking of multimedia content IV*, vol. 5675, pp. 185–196, Feb. 2002.

[85] J.Fridrich, J.Goljan, and R.Du, "Lossless data embedding-new paradigm in digital watermarking," *EURASIP Journal of Applied Signal Processing*, vol. 2002, pp. 185–196, Feb. 2002.

[86] J.Dittman, M.Steinebach, and L.C.Ferri, "Watermarking protocols for authentication and ownership protection based on timestamps and holograms," in *SPIE security and watermarking of multimedia content IV*, no. 4675, pp. 240–251, Jan. 2002.

[87] M.Fallahpour and D.Megías, "Reversible data hiding based on H.264/AVC intra prediction," *Lecture Notes in Computer Science,Springer*, vol. 5450, pp. 52–60, Berlin 2009.

[88] S.K.Kapotas and A.N.Skodras, "Real time data hiding by exploiting the IPCM macroblocks in H.264/AVC streams," *Journal of Real-Time Image Processing*, no. 4, pp. 33–41, 2009.

[89] G.J.Sullivan and T.Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, no. 15, pp. 74–90, 1998.

[90] Y.Hu, C.Zhang, and Y.Su, "Information hiding based on intraprediction modes for H.264/AVC," in *IEEE International Conference on Multimedia and Expo (ICME'07)*, (Beijing,China), pp. 1231–1234, Jul. 2007.

[91] M.A.Ali and E.A.Edirisinghe, "Improved watermark payload capacity using DE on IPCM macroblocks in H.264/AVC," in *5th International Conference on Computer Sciences and Convergence Information Technology (ICCIT'10)*, (Seoul,South Korea), pp. 594–599, Nov.-Dec. 2010.

[92] M.A.Ali and E.A.Edirisinghe, "Multi-layer watermarking of H.264/AVC video using differential expansion on IPCM blocks," in *IEEE International Conference on Consumer Electronics (ICCE'11)*, (Las Vegas,Nevada,USA), pp. 53–54, Jan. 2011.

[93] M.A.Ali and E.A.Edirisinghe, "Reversible watermarking using differential expansion on IPCM macroblocks in H.264/AVC," *JNIT: Journal of Next Generation Information Technology*, vol. 2, no. 1, pp. 105–116, 2011.

[94] M.A.Ali and E.A.Edirisinghe, "Multi-layer reversible watermarking for H.264/AVC video." Submitted to *Signal Processing:Image Communication at Elseiver* [Unpublished], May 2011.

[95] S.Lee and C.D.Yoo, "Robust video fingerprinting for content-based video identification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, pp. 983–988, Jul. 2008.

[96] J. Oostveen, T.Kalker, and J.Haitsma, "Feature extraction and a database strategy for video fingerprinting," in *5th International Conference on Recent Advances in Visual Information Systems*, pp. 117–128, 2002.

[97] K.Iwamoto, E.Kasutani, and A.Yamada, "Image signature robust to caption superimposition for video sequence identification," in *IEEE International Conference on Image Processing (ICIP'06)*, pp. 3185–3188, 8-11 Oct. 2006.

[98] A.Hampapur and R.Bolle, "Feature based indexing for media tracking," in *IEEE International Conference on Multimedia and Expo (ICME '00)*, vol. 3, pp. 1709–1712, 2000.

[99] R.Lienhart, C.Kuhmunch, and W.Effelsberg, "On the detection and recognition of television commercials," in *IEEE International Conference on Multimedia Computing and Systems '97*, pp. 509–516, 3-6 Jun 1997.

[100] J. Sanchez, X. Binefa, and P.Radeva, "Local color analysis for scene break detection applied to TV commercials recognition," in *Visual 99*, pp. 237–244, Jun 1999.

[101] P.Indyk, G.Iyengar, and N.Shivakumar, "Finding pirated video sequences on the internet," technical report, Stanford Infolab, 1999.

[102] R.Radhakrishnan and C.Bauer, "Content-based video signatures based on projections of difference images," in *IEEE 9th Workshop on Multimedia Signal Processing*, pp. 341–344, 1-3 Oct. 2007.

[103] A. Hampapur, K.-K. Hyun, and R.M.Bolle, "Comparison of sequence matching techniques for video copy detection," in *SPIE Storage and Retrieval for Media Databases*, vol. 4676, pp. 194–201, Jan. 2002.

[104] T.Wiegand, G.J.Sullivan, G.Bjontegaard, and A.Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 560–576, Jul. 2003.

[105] D.Alfonso, B.Biffi, and L.Pezzoni, "Adaptive GOP size control in H.264/AVC encoding based on scene change detection," in *Signal Processing Symposium,NORSIG*, pp. 86–89, 7-9 Jun 2006.

[106] "YUV video sequences." Available at: http://trace.eas.asu.edu/yuv/.

[107] R.Mohan, "Video sequence matching," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'98)*, vol. 6, pp. 3697–3700, 12-15 May 1998.

[108] M.A.Ali and E.A.Edirisinghe, "Efficient spatiotemporal matching for video copy detection in H.264/AVC video." Submitted to *IEEE Transactions on Multimedia* [Unpublished], May 2011.

[109] M.Noorkami and R.M.Merserau, "Video watermark detection with controllable performance with and without knowledge of watermark location," in *Multimedia and Security*, pp. 229–236, 2007.

[110] M. Noorkami and R. Mersereau, “Digital video watermarking in P-frames with controlled video bit-rate increase,” *IEEE Transactions on Information Forensics and Security*, vol. 3, pp. 451–455, Sept. 2008.