Some studies on Value Added Services for Connected TV

A thesis submitted for the degree of
Doctor of Philosophy of Jadavpur University

by
Tanushyam Chattopadhyay
Department of Computer Science and Engineering
Faculty of Engineering and Technology
2011

Dedication

Under the feet of the Holy Mother, Sri Sri Sarada Devi.

Acknowledgements

At the very beginning, I would like to thank my supervisors Prof. Mita Nasipuri and Dr. Utpal Garain for their advices, guidance, and insistence on being meticulous in different aspects. I would also like to thank Mr. Debasis Bandyopadhyay, Mr. Arpan Pal and Mr. Aniruddha Sinha for their help and support in different aspects without which it would become impossible to complete my thesis while working in TCS. Also, I thank all HR personnel of CTO team of TCS, who provided me all possible help to complete my work.

I express gratitude towards my colleagues like Mr. Ayan Chaki, Mr. Kaustav Goswami, Mr. Chirabrata Bhaumik and some ex colleagues like Mr. Supriyo Palit, Mr. Anirban Dutta Chowdhury, and Mr. Saurabh Bhattacharya with whom I have had frequent discussion on several aspects of the work embodied in this thesis. Mr. Nihar Chowdhury, Ms. Soumali Roy Chowdhury, Mr. Sounak Dey, Mrs. Pritha Bhattacharya, Mr. Debabrata Pradhan, Mr. Chalamala S. Rao, Mr. Ruchir Gulati helped me a lot for machine implementation of several modules proposed in the thesis. Also, they are my co-authors for some of the papers published/communicated based on the work described in this thesis. I thank them all.

I reserve special thanks for Dr. Pabitra Mitra of IIT Kharagpur, Prof. Bhabatosh Chanda and Prof. Dipti Prasad Mukherjee of Indian statistical Institute to help me in thinking of many ideas. I would also like to thank Dr. Hiranmay Ghosh and Mr. Sitaram Chamarty of TCS for their help and support during my course of work in TCS.

My friends and colleagues at the Innovation Lab, Kolkata of Tata Consultancy Services helped me in many ways in the course of my work. Many of the trainees who worked under my guidance in TCS helped me a lot by providing useful references, generating the corpus content. My special thanks go to Ms. Priyanka Sinha, Mr. Anuran Chattaraj for providing their support for scrutinizing the thesis for minute typographical mistake with utmost care.

I express gratitude towards my parents whose devotion and motivation helped me to reach

into this position of submitting the thesis. I express gratitude towards members of my family - my brother, sister in-laws, and others, for their help and encouragement. I reserve special acknowledgement for my wife, Nilasree, for her great support, cooperation, mental support and inspiration during the entire phase of the research. The same goes for my little son, Arhat.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Introduction

Recent trends on emerging market [1], [2] shows that the connected TV is becoming very popular in developing countries like India, Philippines. Connected TV can be described as an Internet enabled TV. One such product, referred as Home Infotainment Platform (HIP) [3], combines the functions of a television and a computer, by allowing customers to use their television sets for low-bandwidth video chats and access internet websites. It is now commercially available in India [4] and Philippines [5]. The research presented in this thesis is motivated by the business need to implement some value added services, hereinafter referred as **VAS**, on top of this product.

The main reason behind the popularity of this product (HIP) in developing countries is that the Internet penetration in those countries is significantly low [6] compared to the penetration of Television. As per the report [7], 75% of the population of India owns a TV. On the other hand, nowadays many services are hosted in Internet. So with such a low Internet penetration among big section of population exacerbates the digital divide. It is also observed that the low internet penetration is, up to a certain extent due to low household PC penetration not because of the fact that people are reluctant about Internet. Though a significant majority of them have a Television in the home, the potential of interactive television largely remains unfulfilled [8]. As a possible solution, people can browse Internet from the TV using a suitable Set Top Box or other enabling devices [9].

However, the total media viewing and sharing experience is changing and getting richer everyday, as videos, music and other multimedia content flood the Internet. TV has been a favored device of home infotainment for decades. In order to provide an unified Internet Experience on TV, it is imperative that the Internet experience blends into the TV experience. This in turn means that it is necessary to create novel VAS that enrich the standard broadcast TV watching experience through Internet capability. This necessity is eventually translated into the need for

different applications like secured distribution of multimedia content, communication using video chat over TV, and applications that can understand what the user is watching on broadcast TV (referred to as TV-context) and provide user with additional information/interactivity on the same context using the Internet Connectivity. Understanding the basic TV context is quite simple for digital TV broadcast (cable or satellite) using metadata provided in the digital TV stream. But in developing countries, digital TV penetration is quite low. For example, in India, more than 90% TV households still have analog broadcast cable TV. Understanding the TV context in the analog broadcast scenario is really a big challenge. Even for the small percentage of homes where satellite TV has penetrated in form of Direct-to-Home (DTH) service, almost all of them lack in back-channel connectivity for proving true interactivity.

Some of such planned VAS for those connected TVs are video conferencing, Video encryption and watermarking, context based web and TV mash up, video summarization and Electronic Program Guide (EPG) for cable feed channels [10]. This thesis is devoted to the development of those above mentioned VAS for connected TV. As all these services need to be developed on an embedded platform, the primary task is to realize the required video CODEC on the target DSP platform. As H.264 is adjudged as the best video CODEC of the day [11] because of its compression efficiency, video quality, and network friendliness, we have developed most of the VAS on top of H.264 CODEC. Security can be ensured by either encrypting the video or putting a watermark in the video. Context can be extracted at top level by recognizing the channel the user is viewing and then getting the relevant information from the website of that particular channel. On the other hand, textual information in a TV show provide some information related to the show at any particular instant of time.

The major challenge of developing such services in a embedded platform is the resource constraint. The CPU speed and memory size is other constraint as the target hardware determines the cost of the total product.

1.1 Review of Related Works

1.1.1 Realization of Video CODEC on Embedded Platform

In the literature [11] it is reported that the improvement in video quality and compression ratio for H.264 is obtained at the cost of increase in computational complexity. Moreover the memory requirement for implementing the new features of H.264, like multiple frame referencing,

storing the interpolated pixel buffer for half pel and quarter pel prediction, is also very high. Another problem of embedded realization of H.264 is to access the external memory data during executing the code. So the major challenge of realizing H.264 encoder on a DSP platform is in optimizing the computational complexity and memory requirement. So the State of the art is analyzed in light of these challenges.

1.1.1.1 Reduction of Computational Complexity

Two different approaches were taken to reduce the computational complexity which can be measured in terms of Mega Cycles Per Second (MCPS). These approaches are (i)Platform independent optimization and (ii) Platform specific optimization. Some approaches are also there who had described the optimization of the encoder execution time as a whole like [12], [13].

Platform independent optimization: Platform independent optimization techniques are mainly focused in optimization the H.264 encoder in algorithmic level and C/C++ programming level. A number of works can be found on different platform independent optimization techniques for different critical modules of the encoder like [14]. Ivanov et al [15] has shown that the most computationally complex components are Mode Decision and Motion Estimation. In [16], [17], [18] and [19] 'early termination' (ET) and 'forward SKIP prediction' techniques are proposed to optimize the Mode decision module. These techniques are based on the assumption that some block modes can be eliminated from the mode search without any loss, and that correct SKIP decisions may be made at the start of the mode decision making process. These techniques use fast and efficient decision metrics. In [20] optimization is proposed based on dynamic control of the encoding parameters to meet real-time constraints while minimizing coding efficiency loss. In [21] C/C++ level optimization and hardware optimization techniques like intrinsic operator and memory management for C6000 is proposed. In [22] authors have proposed both algorithmic and architectural optimizations of H.264 encoder. In algorithmic optimization they proposed some sub-optimal solution. For example, they suggested 4-Tap FIR filter for half pel prediction instead of 6 Tap FIR filter suggested in H.264 standard.

Platform specific optimization: Some works for optimization on target hardware platform can be found in [21] - [27]. In [25] Wei et. al. discusses a real-time H.264 implementation on a TMS320C6416 DSP. In [27] the hardware system architecture and scheduling is proposed.

1.1.1.2 Memory Optimization

In [28] authors have proposed a novel near-optimal filtering order so that significant reduction in memory requirement is achieved. This work also gives significant reduction reduces MCPS. However, their methodology is applicable to an FPGA prototype. It cannot be used in a commercially available DSP platform, where the user does not have the flexibility to modify the hardware architecture.

The above state of the art reveals the following limitations, too. For example the platform independent optimization techniques gives good optimization but at the cost of coding efficiency. Moreover some of these algorithms are sub optimal and not compliant to the standard. These algorithms are generic and thus can be applied to any type of videos. As the target application is mainly video telephone and video conference, the motion in the videos are very less. So if the nature of video can be exploited, more optimization can be obtained. Thus in this thesis an optimization technique is proposed based on the statistical analysis of the selected mode for these type of videos. On the other hand, the platform specific optimizations are not suited for our choice of video conferencing/video telephone. No literature also focuses on efficient rate control at low cost to get a better video quality.

1.1.2 Video Security

A comprehensive survey on the H.264 video security is described in [29], [30]. As the video security itself is a vast field of research, we have restricted the State of the Art analysis to the study of encryption and watermarking for videos and more specifically for H.264 compressed domain videos.

1.1.2.1 Encryption

Video encryption technique was deployed in the proposed system to stop the illegal distribution of video content. Personal Video Recorder (PVR) enabled Set Top Boxes (STB) now can store the broadcast video content. So the encryption algorithm should satisfy these four following criteria: (i)Robustness, (ii) Compatibility to the video format in which the video would be stored in the PVR, and (iii) Portability so that the proposed method can run on target hardware platform and (iv) Low overhead on coding efficiency. As the supported video format for PVR in the proposed system is H.264, there is a need for robust and low complexity video encryption compatible to H.264 video encoder.

Description of video encryption can be found in [31] - [37]. Some of such techniques are (i) Encrypt the motion vector, (ii) Encrypt the entropy coded stream or (iii) Scramble the prediction modes to achieve encryption. But to the best of our knowledge only few literature is available depicting H.264 based video encryption method. Two such work can be found in [32], [33].

In [32], Yuan Li et. al uses the method of scrambling the intra-prediction, inter-prediction mode and encryption of transform result, motion vectors. Their results show that the proposed method exhibits nice security, has low impact on compression ratio. But the problem with their method is that there is a overhead of keeping a decision block that controls the prediction modes and motion vectors. In [32] Yuanzhi Zou et. al developed their algorithm based on the analysis of H.264 entropy coding, and the algorithm has the features of being irrelative to individual ciphers and adaptive to digital right management. Their scheme applies partial encryption of slice data to preserve the network-friendliness and compression performance of video coding. Beauty of their scheme is that it can efficiently overcome the problem of information leakage resulting from error concealment tools of video coding, and can greatly reduce the hardware design complexity of decryption process, and still retain their virtues. Their algorithm works well for video storage. However, the approach has no negative effect as far as compression ratio and video quality are concerned. But the method is not very robust.

1.1.2.2 Watermarking

Different classifications for watermarking technology is described in [30]. Broadly video watermarking techniques can be classified in two types of approaches namely (i) Pixel Domain where watermark can be directly inserted in the raw video data and (ii) Compressed domain where watermark is integrated during encoding process or implemented by partially decoding the compressed video data. The major problem of implementing the pixel based approaches in the proposed solution is that there is an additional overhead of decoding the compressed video. So it is difficult to meet the real time criteria for those approaches. Moreover the watermarking technique for the proposed system should be compliant to compressed H.264 video format which differs from the previous video codecs in different aspects as described in [11]. We have also described the differences between H.264 and other video codecs in chapter 2.

Now we shall briefly discuss some common video watermarking techniques. Video watermarking techniques involves different approaches with different complexity and robustness. In [38],

[39] Hartung and Girod use Spread Spectrum approach and add an additive watermark into video. Advantage of their method is that it takes care of the artifacts, introduced due to embedding the watermark, by applying drift compensation scheme. In [40],[41] Cox et al uses spread spectrum-based approach for video watermarking, too. In [42] Jordan et al proposed the method of watermark embedding in motion vector. This method of watermarking is of very low complexity. But the problem with this approach is that it may produce artifacts if the video consists of high speed. Also it is not very robust against transcodec attack. In [43], [44] the authors present the video watermarking technique where they modify the middle frequency DCT coefficients: spatially for I frames and temporally for P and B frames. It is an extension of the image-watermarking scheme in [45]. In [46] Langelaar et al presents two different schemes: data hiding by adding labels in MPEG-1 and MPEG-2 bitstream by replacing VLCs of DCT coefficients and Differential Energy Watermarking (DEW) method to embed watermark in video sequence by selectively discarding high frequency coefficients in certain video regions. In [47] Tu et al improve the DEW method by employing energy compensation and setting a threshold for cut-off point. In [48], [49] Swanson et al proposed a multiscale watermarking method working on uncompressed video. This method is based on wavelet transformation and Human Vision Psychology (HVS). This method is very complex. In [50] Linnartz et al proposed the watermarking method based on GOP structure of MPEG-2. In [51] Darmstaedter et al embeds watermark in spatial domain low-pass spread spectrum into 8x8 pixel blocks of video sequence. In [52] Dittmann et al proposes a method in which they decompress the video sequence and treats the video sequence as a collection of images and add the watermark in image and subsequently recompress the image. In [53] Deguillaume et. al embeds watermark in spread spectrum into 3D blocks of video using 3-D DFT. In [54] Busch et al embeds watermark in luminance component of uncompressed video. In [55] Kalker et al uses Just Another watermarking system (JAWS) and embed watermark in spatial domain before compression or after decompression. In [56] Thiemert et al embeds a watermarking bit by enforcing a relationship into group of blocks. In [57] Alattar et al extends the watermarking method by Hartung for MPEG-2 to MPEG-4 bitstream. In [58] Simitopoulos et al alters only the quantized AC coefficients of luminance blocks that belong in intra (I) frames. In [59] Bijan et al embeds watermark in raw stream using spread spectrum method and proves that the watermark is preserved even after decoding using any MPEG-2 decoder. So there remains the challenge of implementing a realtime video watermarking technique for H.264 that is blind, robust and can handle the integrity issue.

1.1.2.3 Watermark Evaluation

Any watermarking scheme can be evaluated by its performance measured in terms of its complexity and robustness against attacks. Any attack to watermarking system is the technique to remove or change the hidden data in the video bitstream. Some of the attacking techniques are listed in [60] to [76].

So in the study of watermarking, it is essential to concentrate on attacks on watermarking scheme, too. Because it is used to simulate the process of any end user trying to remove or destroy the hidden information embedded into the video stream as the security measure taken during watermark embedding. Moreover it is also a measure of goodness of watermarking scheme in terms of robustness of the scheme. But most of the papers on watermarking scheme [77] to [85], don't clearly tell about its robustness against attacks.

1.1.3 Text Information Extraction from Video

The input video format for the proposed system is different for different sources of input signal. The input video may come from a Direct To Home (DTH) service or in form of Radio Frequency (RF) cable. In case of DTH, the input video is in H.264 or MPEG or any other compressed digital video format and on the other hand in case of video RF cable, the input is an analog video signal. In the second case initially the video is digitized for further processing. The Text Information Extraction (TIE) module localizes the candidate text regions from the video. The state of the art shows that the approaches for TIE can be classified broadly in two sets of solution: (i) Using pixel domain information when the input video is in raw format and (ii) Using the compressed domain information when the input video is in compressed format. We shall give a brief overview of both type of approaches as the input to the system can also be either raw or compressed depending on the service provider namely RF cable or DTH respectively. A comprehensive survey on TIE is described in the paper [86] where all different techniques in the literature between 1994 and 2004 have been discussed. So the works between 2005 and 2010 are discussed here.

1.1.3.1 Using the Pixel Domain Video Information

The [86] paper classifies the pixel domain TIE in two classes namely (i) Region based (RB) and (ii) Texture based (TB) approach. TB can be classified into two classes namely (i) Connected component based (CC) and (ii) Edge based (EB) approach. But most of the recent works are based

on texture based approach or edge based approach. It is also observed that the authors prefer to use different features followed by a classifier to localize text regions. A comprehensive survey on the recent work is presented below.

TB approach: In [87] 12 wavelet based features are given as input to some classifier to recognize TIE. In [88] a gradient difference based texture is used to localize the graphics and scene text from the video. Moreover they have applied a novel zero crossing technique that outperforms the projection profile based technique to identify the bounding boxes. In [89] authors have used wavelet transform, statistical features and central moments for detection of both graphics and scene text. They have used projection profile and heuristics to reduce the false positives. In [90] structure elements are used to localize texts from binarized printed documents. [91] have used local binary patterns to extract text features followed by a polynomial neural network (PNN) for classification. [92] have used a sliding window over the frame to extract hybrid features followed by a SVM classifier to localize text regions. They have used morphological operations and vote mechanism to reduce false positives. In [93] authors have used Spacial Auto Correlation (SPAC) method to determine the degree of texture rough details.

EB approaches: In [94] authors have proposed an approach based on background complexities. They presented an iterative method to remove non textual regions based on edge densities. The method described in [95] have used edge based features with minimum computational complexity to localize the text regions so that there is no false negative. They have used SVM to minimize the false positives in turn. The paper [96], [97], [98] have used edge based features and heuristic based filters to localize text regions and applied some geometry based heuristics to reduce the false positives.

Combined feature based approach: [99] have used degree of texture rough-detail to localize the candidate text regions and subsequently used Sobel edge operator to minimize the false positives. [100] have used both texture and edge based approach to localize the candidate text regions and then they have applied SVM to segregate background from foreground.

1.1.3.2 Using the Compressed Domain Video Information

Though there is a lot of work has been done on pixel domain TIE, there is only a little amount of work can be found on TIE when the input is a compressed video. [101] and [102] uses DCT coefficients of I frames and number of Macroblocks (MB) decoded as Intra in a P or B frame for

TIE. [103] uses the horizontal and vertical DCT texture to localize TIE and refine the candidate text regions respectively. [104] uses DCT texture of compressed video to detect TIE. They have considered diagonal edges to handle the Asian Languages like Chinese, Japanese which is not taken care in other papers. They have also used Foreground (FG) background (BG) integrated video text segmentation method to make it robust even in case of complex BG. [105] also used horizontal and vertical energy computed from the DCT coefficients of MPEG stream for TIE. They have used Morphological operations in post processing. [106] used DCT texture followed by a 3x3 median filter in spatial domain for TIE. They have also used some heuristic that the text must reside for at least 2 seconds to remove the false positives. In [107] have used DCT features to extract the text region from MPEG video.

Now the state of the art clearly reveals that a huge amount of research is going on to detect text regions from a streamed video. But the problem with the existing solutions is that they are not considering H.264 as an input video format which is coming to market as a future generation video Codec. Moreover the compressed domain features are more or less based on the texture property of the video. They didn't consider the edge based features that gives good result in pixel domain. Over and above the accuracy of compressed domain approaches are not as accurate as those obtained from pixel domain approach. But the novelty of the compress domain TIE is that they are computationally very efficient.

1.1.4 EPG for RF Enabled TV

[108] describes a method for capturing broadcast EPG data for program title display. However, the detection mechanism of TV channel uses a second tuner in the same television system. The EPG data does not come from the web in the proposed application, rather EPG info is sent over a separate analog channel. The system requires prior training on a defined channel set.

There is not many studies available in literature on systems that identifies the channel from the analog TV video using channel logo and then fetches the relevant EPG information from the Internet. Some related work on the channel logo recognition can be found in [109], [110], [111], [112]. The best performance is observed in x86 platform for the approaches described in [112]. But the approaches taken in [112] involve PCA and ICA which is very much computationally expensive and thus is difficult to be realized in the said DSP platform to get a real time performance. So there is no solution is available in the literature that can recognize the channel logos realtime and

can provide the EPG for RF feed TVs.

1.2 Motivation for the Present Work

The product HIP [4] makes a noise in the Consumer Electronics (CE) market of the developing countries because of its ability to provide internet in TV. But it is always mandatory to provide VAS which may act as a market distinguisher to survive any product in the market. At the same time a lot of research is done on multimedia security and contextual information extraction from video during the last two decades. But all these efforts are made on a PC based platform only. On the other hand, review of the previous works on VAS for HIP like systems reveals that most of the studies concentrate on different sub-problems instead of providing a complete end to end solution. The work embodied in this thesis is motivated to fill this gap.

The major challenge in developing such a product is the resource constraint namely CPU speed and memory of the target hardware. Some of these algorithms describe a good solution for some of the sub-problems in PC environment. But these solutions cannot be implemented on a fixed point DSP platform. The proposed study is focused on developing the following VAS like (i) security of the broadcast video, (ii) context information extraction from streamed video. Moreover all of those solutions need to be deployed in the target hardware. So we have a plan to incorporate the following VAS as a feature for the HIP.

Low bandwidth video applications: This feature enables the user to do video conferencing with another person having the similar HIP installed in his/her home while watching TV. The basic motivation behind this feature is that the TV screen would be minimized to a lower resolution and the user can use the rest part of the TV screen for video conferencing. This feature comes as the wish list from the customers of urban area of India whose wards are working abroad. Another such solution based on low bandwidth requirement is place shifting solution. This solution enables the user to access the home video content over broadband. But both of these solutions can be implemented when there is an efficient video CODEC, satisfying the requirement of high video quality at a low bandwidth, is realized on a DSP platform. As H.264 is proved to be the best video CODEC of the day, we have implemented H.264 on a low cost DSP platform.

Video Encryption: This feature was motivated by the demand from the TV Channel agencies when the PVR was set in the market. The video encryption algorithm allows the user to record the video content using the key which can be derived from the hardware identification number of

the PVR or HIP. As a consequence the user can be tracked if he/she wants to use the recorded content for some illegal business purpose.

Video Watermarking: The need for video watermarking was motivated by the need of one of the major content provider company. They had a need to insert watermark to the content provided by them in a content delivery network (CDN). The same algorithm can be extended for streaming video applications like video on demand (VoD) services provided by the DTH service providers. They also looked for a watermarking evaluation system that can evaluate any watermarking system.

Mash up of TV context and Internet information: Living in a generation of Google TV, Yahoo Connected TV, it is impossible to sustain in the market of connected TV without providing the mash up feature. But as in India most of the people are using analog cable TV and all of those above mentioned products are based on Digital videos only, there is a need to develop such system to address this variation of input, too. Moreover the quality of the videos obtained from analog feed video is quite poor in comparison to those obtained from the DTH service.

EPG for RF feed TVs: The same gap in technology arising from the source of video content motivates us to develop such a service of EPG for the users using RF feed signal for TV.

We have proposed the solutions that can perform at per of the 80% accuracy and efficiency of the related best PC based solution at a 20% cost in terms of execution time and hardware cost. This concept, commonly known as froogle computing, is mainly targeted for the CE products in developing countries. Current thesis is mainly motivated to provide such froogle solutions that can be deployed on the top of the HIP product already developed by the organization.

1.3 Contribution of the Thesis

As far the state of the art is concerned, this thesis has several contributions for development of some VAS for a connected Television. Some of the major contributions are briefly discussed below:

- " At first, the thesis deals with the embedded realization of H.264 video encoder in DSP platform. In this age of multimedia convergence, embedded video based applications for conferencing and streaming have gained big market traction. In this thesis some novel approaches to provide better video quality, coding efficiency and reduced MCPS even under the constraint of target hardware has been proposed. Improvement in video quality and coding efficiency under a constant bit rate is achieved by implementing a novel algorithm for

adaptively selecting the basic unit for rate controlling. The proposed method also reduces the computational complexity using platform independent and platform specific optimization techniques and yet meets the very low memory constraint of the target processor for a standard H.264 baseline encoder without sacrificing the rate-distortion performance. The platform independent optimizations are useful as this version of the code can be ported to any DSP platform for further platform specific optimizations. Almost 40% MCPS reduction with respect to optimized reference code is achieved at the cost of less than 1% reduction in Peak Signal to Noise Ratio (PSNR).

- This thesis deals with an encryption scheme for H.264 video that can be implemented on a DSP platform. In case of Personal Video Recorder (PVR) enabled STBs and connected TVs any user can easily store any TV program. The proposed technique is capable to protect illegal distribution of video content stored in PVR. As H.264 is the preferred storage format for recent PVRs, a method for encryption compatible to H.264 is proposed in this thesis. This thesis presents a fast yet robust video encryption algorithm that performs real-time encryption of the video in H.264 format on a commercially available DSP platform. This algorithm is applied in a real-time place-shifting solution on DSP platform, too. However, the approach has no negative effect as far as compression ratio and video quality are concerned. Mathematically, it can be shown that the proposed method is more robust than those methods for encrypting H.264 video described in the state of the art analysis.

- With the advent of high-speed machine, a hacker, now a day, finds it less difficult to break any encryption key even though it may require large number of attempts. Therefore, an encryption method alone is not sufficient for copyright protection and ownership authentication of stored and streamed videos. In this thesis digital watermarking techniques has been proposed for this purpose. A fast method of watermarking of streamed H.264 video data is proposed in the thesis to meet the real time criteria of a streaming video. This solution was deployed in a content delivery network (CDN) environment, too.

- As different watermarking techniques are evolving, it is extremely required to have a system that can evaluate any watermarking scheme. But to the best of our knowledge there is no significant work to evaluate any video watermarking technique. In this thesis we present a novel technique to evaluate a video watermarking technique by evaluating the video

quality after watermarking and also the robustness of the video watermarking scheme. The proposed technique also compares the retrieved and original watermark (in textual or image format) as a parameter for evaluation.

- In this thesis a novel TV and web mash-up application is described. This application initially extract the relevant textual information from the TV video coming in either analog or digital format and then mash up the related information from the web to provide a true connected TV experience to the viewers. Unlike digital TV transmission it is not possible to automatically get contextual information of TV programs from any Meta data. The text in a TV channel is extracted by text region identification followed by pre-processing of the text regions and performing Optical Character Recognition (OCR) on the text regions. The applications are presented for x86 based PC platform and ARM based dual-core platform. This type of system is not available in the literature.

- The thesis presents a novel method for recognizing the channel logos from the streamed videos in real time, which has various applications for VAS in the connected TV space. The prototype is developed in X86 platform and then ported on a commercially available DSP with nearly 100% accuracy in real time. In India, where most of the people are still watching TV using Radio Frequency (RF) feed cable, this image processing based approach solution for providing EPG is novel in nature.

1.4 Organization of the Thesis

The content of the thesis can be broadly divided into three major parts:

(i) *Realization of video CODEC on target DSP platform*: Chapter 2 deals with the platform independent and platform specific optimizations of the encoder to run in realtime in the target hardware;

(ii) *Security*: Chapter 3 is devoted for this purpose. In this chapter we have discussed about the encryption and watermarking technique compatible to H.264 video CODEC that can be realized on a target DSP platform also.

(iii) *TV context understanding based applications*: Chapter 4 and Chapter 5 deals with these two aspects related to TV context understanding based applications namely (i) Mash up of Textual context of broadcast video and Web Information and (ii)Generation of Electronic Program Guide

for RF fed connected Televisions. Chapter 6 concludes the thesis by discussing about the goals of the thesis, achievement and the future scope of research.

A chapter-wise break up of the thesis is briefly given below:

Chapter 2. The current thesis is dedicated to Development of VAS for connected TVs. To provide these services the first step is to implement some video encoder which is required to get the video in a compressed format and that encoder must be realized/implemented on a suitable DSP platform that can support all the proposed VAS at an optimum cost. The coding efficiency of any video encoder is usually judged by the rate distortion curve where the video quality in PSNR is plotted against the bit rate requirement. Putting it in other word, the best video encoder is one that can give best video quality with minimum storage requirement. Considering this criteria H.264 is proved to be the best video encoder. But it is also true that this coding efficiency of H.264 is achieved at the cost of coding complexity. So it is difficult to run on a DSP platform. So our first chapter is devoted to realization of H.264 on DSP platform.

Chapter 3. While the TV shows are recorded into storage, the legal issue comes with the copyright of these contents. It is quite likely that any customer of such a system can illegally sell the contents after recording it into his/her personal device. So some suitable method to protect the copyright should be implemented along with the video encoder so that this Intellectual Property Right (IPR) related issues don't arise. One way to do this is to apply some encryption and watermarking techniques to ensure the security of the multimedia content streamed by the channels. Moreover this security should be implemented in real time so that the security measure is taken during encoding the video into PVR and no additional overhead is there. So Chapter 3 is dedicated to implementation of security measures in real time on the same target DSP platform and compatible to H.264 video encoder.

Chapter 4 One of the VAS proposed in the thesis is an application based on localization and recognition of text contents within a video in order to provide additional news feed information while showing the news videos. This VAS is based on mainly four major modules namely (i) text localization, (ii) text processing and noise removal for image enhancement, (iii) text recognition, and (iv) keyword spotting. For the recognition part we have used some third party tool and thus, the Chapter 4 discusses about the text localization, preprocessing for recognition, and keyword spotting only. The input video may be in a compressed digital format or in analog format. To extract the text regions efficiently, the thesis discusses about the text localization method in both

compressed domain and pixel domain.

Chapter 5 In this chapter a system for showing EPG information even in case of RF fed TV is described. This system works by recognizing the channel by identifying the logo of the channel. After the channel is identified, the EPG is obtained by looking up the website of the particular channel. The EPG is finally rendered on the video of the channel. The system is developed on a DSP platform that can recognize the channel within a delay of 2 seconds after changing the channel.

Chapter 6 In this chapter we conclude the thesis by initially discussing the goals we have set at the beginning and then discuss how these goals were achieved. Finally we have discussed the scope of future research.

Chapter 2

Realization of H.264 Video Encoder on Low Cost DSP Platforms

2.1 Introduction

One of the Value Added Services (VAS) for a connected TV proposed in this thesis is a point to point video conference. Two basic modules for implementing such a video conference service are video encoder and decoder. Video encoder is required to compress the video captured by the web cam attached to the system so that the video can be efficiently transmitted over internet. Video decoder is required to decode the compressed video sent from the other end of video conference. As the video decoder is very much standard bound, there is hardly any scope of research. On the other hand video encoder is an open ended system and as a thumb rule three times more complex than the decoder. Moreover there is a need for a video encoder for some other VAS like storing the streamed video in PVR, place shifting application. So all of those services, there is a need to implement a video encoder that can encode the video in real time with high compression. So two major challenges namely (i) selection of a suitable video encoder and (ii) realization of that video encoder on target Digital Signal Processors (DSP) platform are there.

Several video CODECs have been developed since mid 80's of the last century. The coding efficiency of any video encoder is usually judged by rate distortion curve where the video quality (in PSNR) is plotted against the bit rate requirement. Considering this criteria H.264 is proved to be the best video encoder. H.264 also gives better coding efficiency, better video quality and network friendliness [113] - [115]. But these improvements are achieved at the cost of higher computational complexity.

Therefore when a H.264 encoder needs to be implemented on a DSP platform, it becomes a challenging problem for the researcher in Consumer Electronics. Any embedded implementation requires optimization of different parameters like cost of the processor, which also depends on

processor architecture, speed, and memory. Therefore to make an attractive consumer electronics product it is always better to select a low cost processor which can serve the user requirements. But such processors have a limited resource in terms of memory and CPU speed. So implementing a video encoder on such a platform requires some algorithmic and platform specific improvements so that a good video quality (measured in terms of PSNR) can be obtained at a significantly low bitrate in real time. For real time application, computational complexity also becomes an important parameter as it affects both power consumption and real time performance. One also needs to cater for the memory constraint of the target hardware platform.

In this chapter we propose some approach to implementing video encoders that improves the video quality, coding efficiency and reduces the computational cost measured in terms of Mega Cycles Per Second (MCPS) even under the constraint of target hardware. Improvement in video quality and coding efficiency under a constant bit rate is achieved by implementing an algorithm for selecting the basic unit for rate controlling adaptively (in Macroblock level or frame level). In this chapter we have also described the method to reduce the computational complexity using platform independent and platform specific optimization techniques and yet meets the very low memory constraint of the target processor for a standard H.264 baseline encoder without sacrificing the rate-distortion performance.

The proposed algorithm is tested against ITU-T test sequence of different resolutions. Implementation details are presented for a QCIF H.264 baseline encoder on a low cost commercially available DSP platform, which has only 256 KB RAM and 150 MHz clock speed. Results shows that almost 40% MCPS reduction with respect to optimized reference code is achieved at the cost of less than 1% reduction in Peak Signal to Noise Ratio (PSNR).

In this chapter we shall first give an overview of H.264, then describe the complexity analysis of H.264 Encoder, proposed method for algorithm and platform specific optimizations, and finally discuss some use cases of realization of H.264 video CODEC on DSP platform.

2.2 Overview of H.264 Encoder

This section covers the overview of H.264 video Encoder. H.264 video CODEC is the latest video codec standard [113] by ITU-T. The tools and functionality of different modules of H.264 standard are nicely explained in [11], [115]. Compared with MPEG-2, H.264 mainly introduces the following changes: (i) enhancements of the ability to predict the values of the content of a picture

to be encoded, (ii) methods for improving coding efficiency, and (iii) robustness to a variety of network environments. To introduce these changes some tools are added to H.264 which were not there in the previous codecs namely:

- Variable block-size motion compensation with small block sizes,
- Quarter-sample accuracy for motion compensation,
- Motion vectors over picture boundaries,
- Multiple reference picture motion compensation,
- Decoupling of referencing order from display order,
- Decoupling of picture representation methods from the ability to use a picture for reference,
- Weighted prediction,
- Improved “skipped” and “direct” motion inference,
- Directional spatial prediction for intra coding, and
- In loop deblocking filtering.

In addition to improved prediction methods, other aspects of the design were also enhanced for improved coding efficiency, including:

- Small block-size transform,
- Hierarchical block transform,
- Short word-length transform,
- Exact-match transform,
- Arithmetic entropy coding, and
- Context-adaptive entropy coding.

And for robustness to data errors/losses and flexibility for operation over a variety of network environments, some key design aspects include:

- Parameter set structure,
- NAL unit syntax structure,
- Flexible slice size,
- Flexible macroblock ordering (FMO),
- Arbitrary Slice Ordering (ASO),
- Redundant pictures,
- Data partitioning, and
- SP/SI synchronization switching pictures.

To address the large range of applications considered by H.264, three profiles have been defined: namely Baseline Profile, Main Profile, Extended Profile. In the latest standard Fidelity Range Extetension profile (FRExt) is also added. These profiles are defined based on different target applications as described in table 2.1.

Table 2.1: Profile Wise Application Distribution

Application	Requirements	Profile
Broadcast Television	Coding efficiency; reliability; interlace; low complexity decoder	Main
Video Playback	Coding efficiency; interlace; low complexity encoder and decoder	Main
Studio Distribution	Lossless or near lossless; interlace; efficient transcoding	FRExt
Streaming video	Coding efficiency; reliability; scalability	Extended
Video Conferencing	Coding efficiency; reliability; low latency; low complexity encoder and decoder	Baseline
Mobile Video	Coding efficiency; reliability; low latency; low complexity encoder and decoder; low power consumption	Baseline

As the planned VAS for connected TV is video conferencing we have implemented baseline profile of H.264 on the target DSP platform. This baseline profile includes the following tools:

- I and P slices,
- Quarter pixel motion compensation,
- Multiple reference frames,
- In-loop de-blocking filter,
- 9 modes in intra prediction,
- Variable block sizes,
- Flexible macroblock ordering (FMO),
- Redundant pictures,
- Arbitrary slice ordering (ASO),
- Context-Adaptive Variable Length Coding (CAVLC).

The overview of H.264 is described in Figure 2.1. This diagram shows that the basic functional modules of H.264 encoder are Prediction module (Intra and Inter), Transform, Quantization, Rescaling and Inverse Transform module, Deblocking filter, Entropy coding. A brief description of each module is described below:

2.2.1 Intra Prediction

This prediction uses the spatial redundancy to achieve data compression. It is described as a prediction derived from the decoded samples of the same decoded slice.

2.2.2 Inter Prediction

This prediction uses the temporal redundancy among consecutive video frames to achieve the compression. This is described as a prediction derived from decoded samples of reference pictures in the reference picture buffer other than the current decoded picture. It is obvious that better motion compensation can be achieved if motion vector (MV) can take fractional values rather than just integer values as the object in the video not necessarily move by an integer number of pixels from the previous frame to the current frame. Increased fractional accuracy can provide a better match and thus the energy in the motion-compensated residual is reduced. But Sub-pixel motion

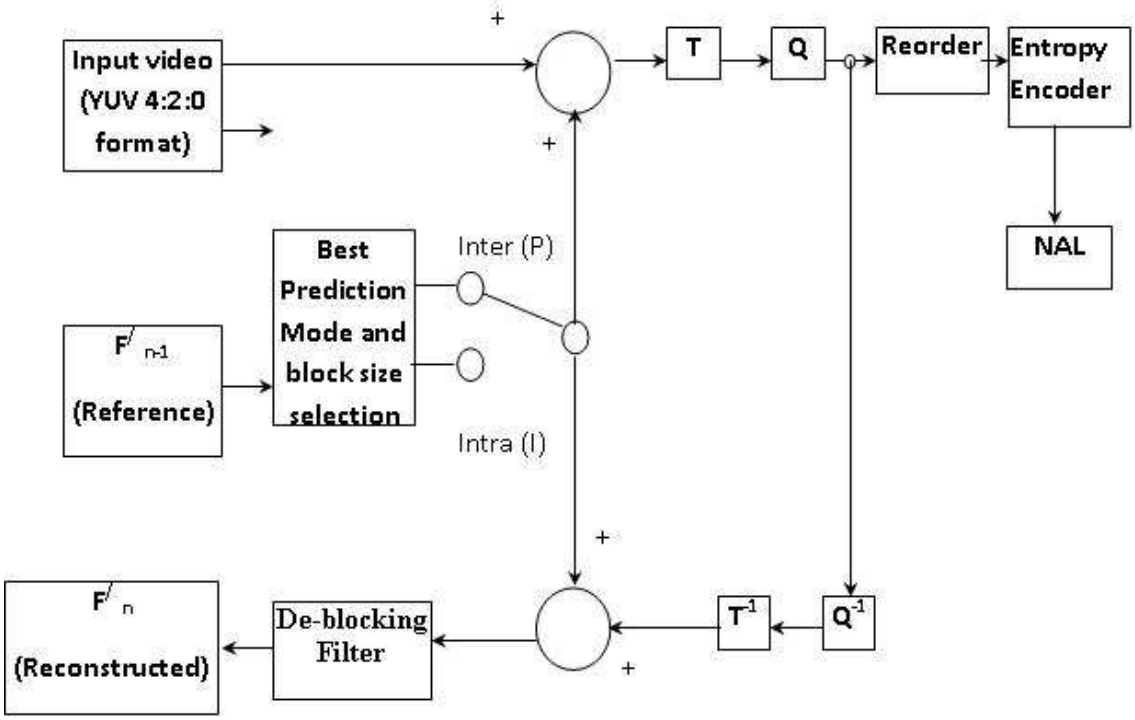

Figure 2.1: Block Diagram of H.264/AVC Encoder

estimation requires the encoder to interpolate between integer sample positions in the reference frame which is very computationally expensive, especially so for quarter-pixel interpolation because a high-order interpolation filter is required for good compression performance. In H.264 half pixels are interpolated using a 6-Tap FIR filter and these values are used to interpolate quarter pixels. Different search techniques are defined to find the motion vector for which the residue energy is minimized.

2.2.3 Transform, Quantization, Rescaling and Inverse Transform

These modules are used for coding the residue part. H.264 uses three transforms depending on the type of residual data that is to be coded: (i) a Hadamard transform for the 4x4 array of luma DC coefficients in intra macroblocks predicted in 16x16 mode, (ii) a Hadamard transform for the 2 x 2 array of chroma DC coefficients (in any macroblock) and (iii) a DCT-based transform for all other 4 x 4 blocks in the residual data.

2.2.4 Deblocking Filter

It is a conditional filter applied to all 4x4 block edges of a picture except edges at the boundary of the picture and any edges for which the filter is disabled to reduce blocking distortion. The deblocking filter is applied after the inverse transform in the encoder and in the decoder. The filter smoothes block edges, improving the appearance of decoded frames. The filtered image is used for motion-compensated prediction of future frames and this can improve compression performance because the filtered image is often a more faithful reproduction of the original frame than a blocky, unfiltered image.

2.2.5 Entropy Coding

Above the slice layer, syntax elements are encoded as fixed- or variable-length binary codes. At the slice layer and below, elements are coded using either variable-length codes (VLCs) or context-adaptive arithmetic coding (CABAC) depending on the entropy encoding mode. But in Baseline profile there is no CABAC and thus we shall consider only VLC in our discussion.

Table 2.2: Estimation of Time Complexity per Basic Operational Units

Basic tool (per macroblock)	MAC	ADD	MPY
Inter prediction	1409	45305	6400
Intra prediction	48	13660	1325
Transformation and inverse transformation	0	224	48
Deblocking filter	0	5376	384

2.3 Complexity Analysis of H.264 Encoder

A comprehensive analysis of computational complexity for H.264 decoder can be found from [116]. However no such proper analysis of time complexity for H.264 encoder can be found from the literature. We shall describe the complexity of H.264 encoder in terms of the complexity of its basic modules described earlier. Among those modules, VLC is more control intensive than computational operations. Therefore, the complexity of H.264 encoder algorithm is described in terms rest of the basic operational units in terms of the major operations like Multiplication and Accumulation (MAC), Addition (ADD), and multiplication (MPY) as in Table 2.2. The distribution of ADD operation in different modules is depicted in Figure 2.2 .

Details of the sub units of those modules are described in Table 2.3. In this module we have placed the interpolation as a separate entry in the table though it is a part of Inter prediction as it is a significantly computationally expensive module.

2.4 Algorithmic Optimizations

Major challenges for implementing H.264 video encoder on DSP platforms are (i) CPU constraint and (ii) Memory constraint. So we have focused in two aspects of optimization namely optimization of MCPS and optimization of memory requirement.

2.4.1 MCPS Optimization

H.264 encoder estimates the cost for all possible intra and inter prediction modes before selecting the best mode i.e. the mode with the least cost. Therefore the mode computation module is the most computationally expensive module in H.264 encoder. Here we have proposed a method of early termination of prediction mode computation based on statistical analysis.

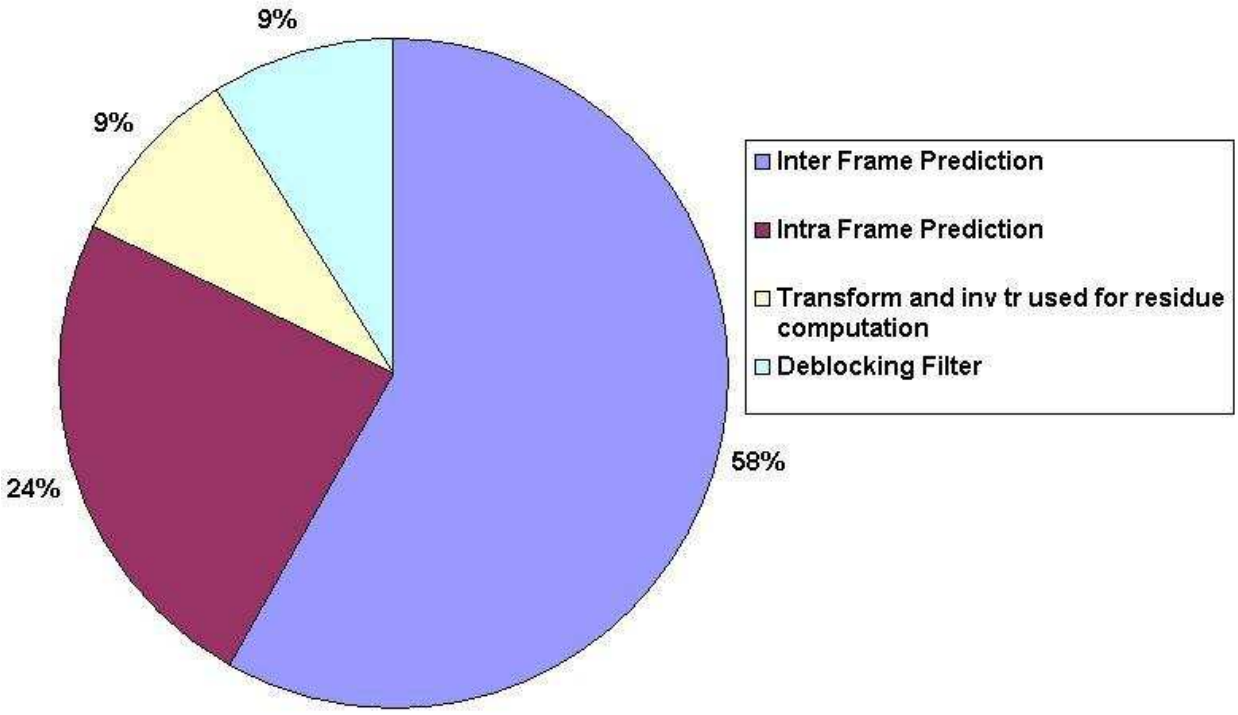

Figure 2.2: π Chart Showing Complexity of Different Sub Modules for ADD Operation

Table 2.3: Estimation of Time Complexity per Module

Basic tool	sub module	MAC	ADD	MPY
Deblocking filer (per Macro Block)	Luma	0	3584	256
	Cb	0	896	64
	Cr	0	896	64
Transformation and inverse transformation (Per macroblock) including Luma AC, Luma DC, Chroma AC and Chroma DC)	Residue computaion	0	16	0
	Integer Transform	0	80	0
	Scaling	0	32	32
	Inverse Integer Transform	0	80	0
	Inv Scaling	0	16	16
Intra prediction (per macroblock)	Luma 4 x 4 (9 modes)	0	4560	544
	Luma 16 x 16 (4 modes and Hadamard used)	16	7414	515
	Chroma (Cb) (All modes and Hadamard used)	16	843	133
	Chroma (Cr) (All modes and Hadamard used)	16	843	133
Inter prediction (per macroblock)	Luma 16 x 16 (1/2 pel and 1/4pel)	337	24105	0
	Luma 4 x4 (1/2 pel and 1/4 pel)	1072	10448	0
	Chroma (Cb) (8x8)	0	384	512
	Chroma (Cr) (8x8)	0	384	512
Interpolation per pixel	Half pel	0	15	9
	Quarter pel	0	24	12

2.4.1.1 Early Termination of Prediction Mode Computation

Inter predicted frames in a H.264 video encoder supports different block sizes, like 16x16, 16x8, 8x16, 8x8, 4x8, 8x4 and 4x4 [11] for partitioning of the macroblock. Smaller block size increases the video quality at the cost of coding efficiency. On the other hand, larger block sizes are preferred for better compression at the cost of image quality. So to achieve some optimization in MCPS without any significant change in coding efficiency or video quality we have selected 16x16 and 4x4 block size out of those 7 possibilities for Inter frame prediction. In the proposed solution Hadamard transformation is used for 16X16 intra prediction but not for 4X4 prediction. So in the proposed realization, for each macroblock (MB), prediction cost for four modes namely Inter 4x4, Inter 16x16, Intra 4x4, and Intra 16x16 are computed for Inter prediction. Similarly for every Intra frame the prediction cost is computed for Intra 4x4, Intra 16x16. The diagram depicting the traditional prediction mode computation is shown in Figure 2.3.

So, if each of these modules takes t time, the total time taken in prediction module will be 4t in a Inter predicted frame. Now if the prediction mode can be predicted correctly it can save up to 3t time per macroblock. Table 2.3 clearly suggests that a huge number of cycles can be saved if the prediction mode can be predicted beforehand. We have assumed that the input video for video conferencing or video phone applications would have very less motion. So we have proposed an early termination technique based on the statistical analysis on prediction cost and selected prediction mode for eight slow moving reference test sequences. These reference test videos were obtained from the ITU-T website. This analysis report is shown in Table 2.4. The statistical analysis shows that most of the time P16x16 prediction mode is selected as the best prediction mode. Our proposed termination method is depicted in Figure 2.4. Inter prediction with block size 4x4, Inter prediction with block size 16x16, Intra prediction with block size 4x4 and Intra prediction with block size 16x16 will be referred to as P4x4, P16x16, I4x4, I16x16 respectively in the subsequent section. The proposed early termination algorithm is a threshold based approach. The method for computing the threshold is given here:

- For all these test sequences the prediction cost for each selected mode is noted. Let $CI_{16}(i,j)$, $CI_4(i,j)$,$CP_4(i,j)$ be the cost of prediction for j^{th} macroblock of i^{th} frame when I16x16, I4x4 and P4x4 is selected as the best prediction mode.

- The average of Cx(i,j) is computed for all i and j where x is I16, I4, P4.

Table 2.4: Statistical Analysis of Selected Mode for Different Streams

Sequence	Total MB	P4MB	P16MB	I4MB	I16MB	No of forced I MB	% of P16MB
Akiyo	8910	25	8588	235	62	297	99.71
News	8910	16	8590	244	60	297	99.73
Claire	8910	28	8554	138	190	297	99.32
grandma	8910	3	8610	227	70	297	99.97
salesman	8910	16	8580	289	25	297	99.62
hall-monitor	8910	43	8545	240	82	297	99.21
Bridge	8910	0	8313	267	330	297	96.52
Container	8910	0	8613	237	60	297	96.66

- These averages are treated as threshold for this particular mode. Let them be defined as TI16, TI4, TP4.
- For all Inter frames, initially prediction cost for P16x16 (CP16) is computed and then the following conditions are checked
- If $CP_{16} > TP4$ go for P4x4 prediction, if $CP_{16} > TI4$ go for I4x4 prediction and $CP_{16} > TI16$ go for I16x16 prediction. Else computation of prediction cost for all other prediction modes is bypassed.

2.4.1.2 Early Skip on MV Computation

From a similar analysis on Motion vector it can be found that the number of sub-blocks with non-zero motion vectors for Akiyo and news sequence is 13.78% and 17.08% respectively. Therefore an early termination method can be formulated based on the following logic:

- Find the sum of absolute difference (SAD) for 0 motion vector.
- If SAD crosses a threshold value T_{mv}, search for other integer pixel motion vectors only.
- If any of these motion vector again leads to a SAD less than T_{mv}then terminate the MV computation process
- Repeat the same process for Sub-pixel (Half pixel and quarter pixel) MV

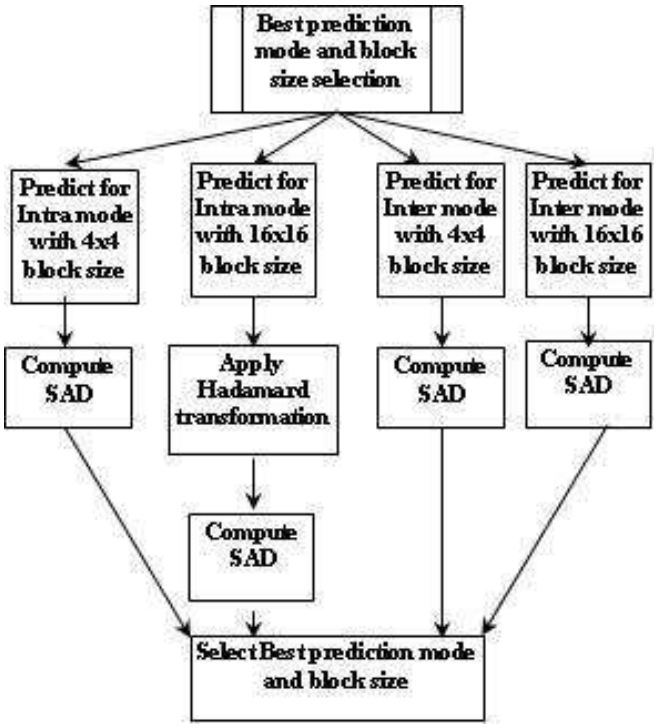

Figure 2.3: Flow Chart of Best Prediction Mode and Block Size Selection in Traditional Design

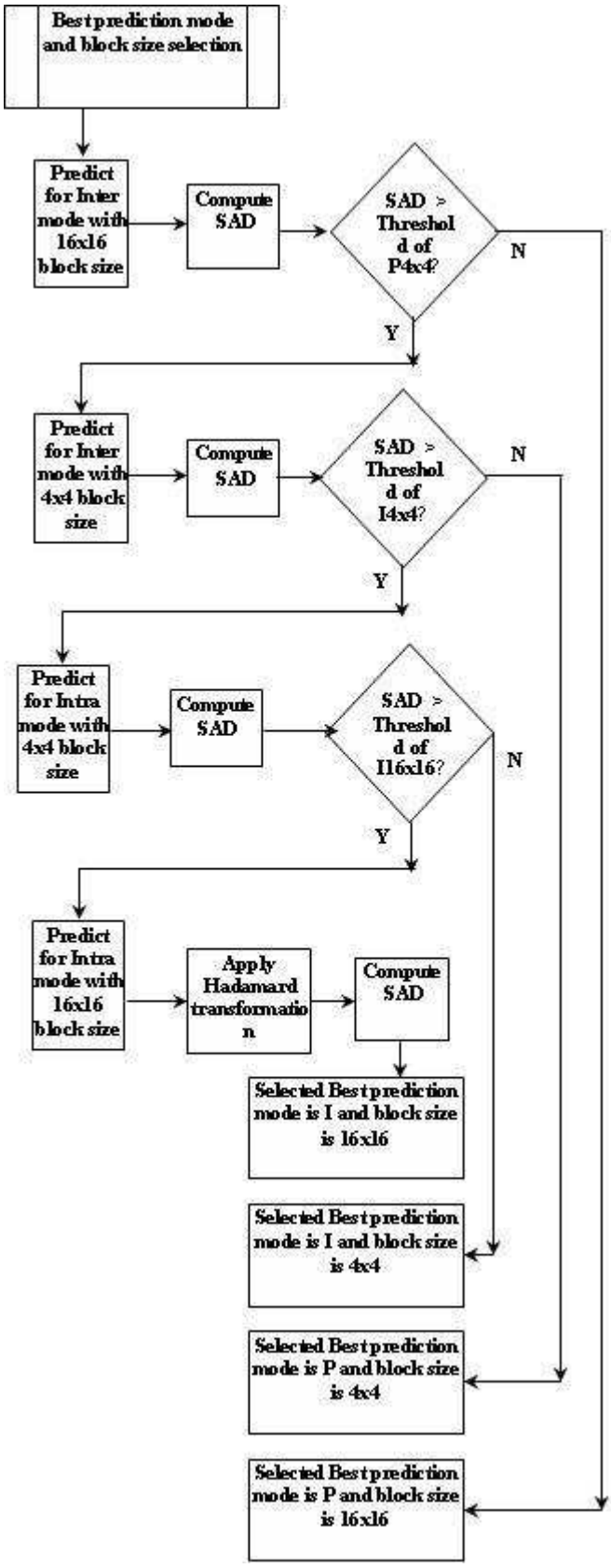

Figure 2.4: Flow Chart of Best Prediction Mode and Block Size Selection in Proposed Algorithm

Table 2.5: Memory Requirement for Different Modules

Buffer name	Formula	QCIF
Half pel of location (2,0)	(W+2*P)*(H+2*P)	28,644
Half pel of location (0,2)	(W+2*P)*(H+2*P)	28,644
Half pel of location (2,2)	(W+2*P)*(H+2*P)	28,644
Temporary Half pel (for computation of location (2,2))	(W+P)*(H+P)	26,969
Quarter Pel	16*16	256

- In this work T_{mv} is computed as 32, as it leads to average residue value per pixel as 2 and after quantization they leads to very little overhead on coding efficiency. This saves the time of search for sub-pixel motion vectors.

2.4.2 Algorithmic Modifications to Meet Memory Constraint

In this section all the assumptions are stated first, then the memory requirement is analyzed and finally the proposed algorithm is described. These optimizations are based on the following assumptions:

- Computational cost for one ADD and one internal memory access is lesser than fetching data from external memory.
- Syntax element buffer, macroblock buffer are not considered in the memory analysis.
- All operations are performed on padded reference buffer to reduce boundary check.
- For slow moving sequences, which are relevant for video conferencing or videophone like applications, quarter pixel prediction occurs in very less frequently. So it is better to compute them on demand than compute it at a time.

Memory requirement analysis for H.264 baseline decoder is described in [116]. However there is no such analysis for H.264 encoder. Therefore a brief memory requirement analysis of the buffers is presented here in Table 2.5. The following notations are used in the memory requirement analysis: W: width, H: Height, P: Size of padding. For QCIF resolution W=176, H=144 and in the proposed system P=5.

The half pixel computation is a quite MCPS hungry process as 6 tap FIR filter is used. On the other hand one half pixel may be referred repeatedly for prediction of different pixels, all half

pixel of a inter frame are interpolated at a time. But as per assumption, quarter pixels are referred less frequently. So the quarter pixels are interpolated on demand for a 16x16 macroblock. In the reference code by HHI all half pixels and quarter pixels are interpolated at a time which requires a buffer of W*H*4*4 size. Moreover the selected DSP platform has minimum data width as 16 bit, so this buffer takes 792 Kbytes for a video frame of QCIF resolution. But as memory is a constraint on the selected DSP platform, three buffers of size W*H for half pixel and one for full pel data are used in the proposed design. This modification in design takes 198Kbytes memory and thus saves memory by 75%.

2.5 Platform Specific Optimizations

On average 43,410,666.4 cycles per frame is required when the code with the above mentioned optimizations are ported into the target DSP platform. So it is evident that platform specific optimizations on the selected DSP platform are required to meet the system requirement. These optimization techniques are described below:

2.5.1 Use of Dual MAC for Half-pel Prediction

This section describes how to use dual MAC in assembly language for Half pel interpolation.

- Use two registers (AR0 and AR4) to store the first and last pointers of pixel value (using which the interpolation needs to be done) buffer respectively
- So initially the AR4 can be obtained by adding 5 to AR0
- Store the filter values of 6-TAP Filter in a register CDP
- Use the register AC3 to accumulate results
- Use dual MAC as:

  MPY *AR0+, *CDP+, AC0

  ::MAC *AR4-, *CDP+, AC3

  MAC *AR0+, *CDP+, AC0

  ::MAC *AR4-, *CDP+, AC3

  MAC *AR0-, *CDP+, AC0

::MAC *(AR4+T0), *CDP+, AC3

This operation will complete the Interpolation using H.264 standard compliant 6-Tap FIR filter.

2.5.2 Use of Intrinsic

The selected DSP platform supports a set of intrinsic like MAX, MIN, ABS. In H.264 encoder boundary check is required when the modules like residue color transformation, de-blocking filter and interpolation are executed. Boundary check can be implemented very efficiently using max and min Intrinsic. Similarly, ABS is used in many functions of VLC, de-blocking filter, and inter and intra prediction. The use of these intrinsic led to optimization.

2.5.3 Use of EDMA

While implementing the H.264 encoder on the selected DSP platform a huge number of cycles are spent in data transfer from external to internal memory. One good option to minimize this is to use EDMA. The data access requirements for each of the encoding modules described in Table 2.6.

2.5.3.1 External Access through DMA

As discussed earlier, inter motion estimation, picture reconstruction and reference frame buffer management modules require frequent external to and fro internal memory access. So, it is important to minimize the delay because of those external memory access during execution of those modules. The motion estimation (ME) and picture reconstruction (PIC) blocks operate on a 4 x 4 Luma component (hereinafter referred as Y) and 2 x 2 Chrominance component (hereinafter referred as UV) macroblock partition whereas the buffer management (BUF) block operates once per frame. The ME and PIC block runs iteratively using a raster scan order. So, for these blocks, (a) the block size is fixed, (b) the external memory address is also well-defined at the start of a macroblock. The buffer size (one video frame) and the target address (reference buffer pointer) is well-defined for the BUF module also. Hence, the ME, PIC and BUF modules can be effectively loaded/stored using DMA (Direct Memory Access). So the BUF module may use a simple DMA engine, but the requirement for the ME and PIC module are complicated as they need DMA access iteratively for every 4 x 4 Y (2 x 2 UV) partition.

Table 2.6: Data Access Summary

Module	Data Access Requirement (Internal $<->$ External)	Remarks
Bitstream Parsing (BP)	None	Bitstream buffer should be placed in internal memory, since H.264 parsing is context dependent, hence it is difficult to fetch a self-contained block of data from the bitstream buffer and parse
Motion Estimation (ME)	Inter: External(Reference Frame) : Internal (Scratch Motion Prediction Buffer) Intra: External (Decoded Picture): Internal (Scratch Motion Prediction Buffer)	Can be effectively handled using DMA, once every 4 x 4 Y (2 x 2 UV) partition Suitable for Cache
Residual Error Computation (RES)	None	Can be computed on the motion prediction buffer or any other scratch buffer
Picture Reconstruction (PIC)	Internal (Scratch Buffer); External (Decoded Picture)	Can be effectively handled using DMA
Deblocking (DB)	External (Decoded Picture) Internal (Scratch Buffer) Internal (Scratch Buffer) External (Decoded Picture)	Suitable for Cache
Buffer Management (BUF)	External (Decoded Picture) External (Reference Frame)	Can be effectively handled using DMA, only once per frame

2.5.3.2 Buffer Management (BUF)

During encoding the $(n+1)^{th}$ frame, the decoded picture for the n^{th} frame is ready for buffer management as well as for display. The decoded picture needs to be pushed to the reference frame stack for future access. Typically the Y component of the decoded picture buffer is padded and stored as reference frame for better accessibility during motion compensation. So the reference frame is padded using CPU and then it is transferred to the active picture using DMA. Both the Y and UV active picture buffers can be activated using two parallel DMA channels (assuming UV component is not padded and is contiguous) and can be transferred to the reference frame at a single transfer. The reference frame is accessed at the ME block in the next frame, so the CPU has to check for completion at that point.

2.5.3.3 Motion Estimation (ME) - Inter Prediction

Motion Estimation for inter frames use previously stored reference frames for motion prediction. Motion prediction can be done either at full pixel resolution or using interpolation for fractional resolution (half/quarter pixel).

For full pixel interpolation, a 4 x 4 Y and 2 x 2 UV is copied directly from reference buffer to scratch motion prediction buffer. For fractional pixel interpolation, 9 x 9 Y and 3 x 3 UV is copied directly from reference buffer to internal scratch motion prediction buffer. This scratch buffer is interpolated to derive the 4 x 4 Y (2 x 2 UV) motion prediction buffer. Note that, for Y, depending on the prediction mode a 9 x 4 or a 4 x 9 block can also be used for interpolation, whereas for UV, a 3 x 2 or a 2 x 3 can also be used for interpolation. Hence, for memory critical requirements, instead of using 9 x 9 Y (3 x 3 UV), one can dynamically use smaller blocks depending on the prediction mode. For the current analysis, we assume a 9 x 9 Y (3 x 3 UV) as the scratch buffer irrespective of the prediction mode.

Any 16 x 16 Y (8 x 8 UV) macroblock consists of 16 4 x 4 Y (2 x 2 UV) sub macroblocks. Now, each of these sub macroblocks requires both (a) 4 x 4 Y (2 x 2 UV) reference block for full pel prediction or (b) 9 x 9 Y (3 x 3 UV) reference block for sub pel prediction. Information regarding (a) and (b) are available during reading the raw input video file. During encoding, file reading is done prior to intra prediction; hence, DMA can be activated for the macroblock in parallel to intra prediction. For the Y component, two types of accesses are required, (a) for 4 x 4 full pel prediction, (b) for 9 x 9 sub pel prediction. These accesses should happen repetitively until all the

external access for the current macroblock is fulfilled. There can be a maximum of 16 (a) type access or 16 (b) type access or a combination of both. For the UV component, two types of accesses are required, (a) for 2 x 2 full pel prediction, (b) for 3 x 3 sub pel prediction. These accesses should happen repetitively until all the external access for the current macroblock is fulfilled. There can be a maximum of 16 (a) type access or 16 (b) type access or a combination of both.

After the motion vector parsing, a Y lookup table can be generated which has 16 entries, each having the following information

- Source Address for Y component in the reference frame (external)
- Destination Address for Y component in internal memory (motion vector predictor buffer for 4 x 4/scratch buffer for 9 x 9)
- Size of the block (4 x 4 for full pel, 9 x 9 for sub pel) Similarly, a U lookup table can be generated which has 16 entries, each having the following information
- Source Address for U component in the reference frame (external)
- Destination Address for U component in internal memory (motion vector predictor buffer for 2 x 2/scratch buffer for 3 x 3)
- Size of the block (2 x 2 for full pel, 3 x 3 for sub pel) Also, a V lookup table can be generated which has 16 entries, each having the following information
- Source Address for V component in the reference frame (external)
- Destination Address for V component in internal memory (motion vector predictor buffer for 2 x 2/scratch buffer for 3 x 3)
- Size of the block (2 x 2 for full pel, 3 x 3 for sub pel)

Once the Y, U and V lookup table has been generated based on motion vector parsing, the DMA channels can be activated. The DMA channels can be activated simultaneously, if there is enough internal memory to hold scratch buffers for Y, U and V simultaneously. Otherwise, the DMA channels can be activated sequentially. The scratch memory requirements for two different scenarios namely high and low scratch memory is shown in Table 2.7 and Table 2.8. Note that, 16 buffers for 4 x 4 (2 x 2) and 9 x 9 (3 x 3) need to be allocated before hand, since it is not known

Table 2.7: Scratch Memory (High) for Motion Estimation - I

Colour Component	Number of buffers	Total (bytes)
Y (4 x 4)	16	256
Y (9 x 9)	16	1296
U (2 x 2)	16	64
U (3 x 3)	16	144
V (2 x 2)	16	64
V (3 x 3)	16	144
	Total	1968

which of the buffers will be used during motion compensation. However, in this scheme, there is a significant amount of memory wastage. For an optimal memory design, one can embed the 4 x 4 buffers within the 9 x 9 buffers and manipulate pointers accordingly.

2.6 Enhancements in Video Quality under a Fixed Bit Rate

2.6.1 Use of Adaptive Basic Unit Selection for Rate Controlling

According to [11], quantization parameter may vary at the frame level, slice level or macroblock level. Similarly, in [119] rate controlling in different basic units is defined. A larger basic unit size increases bit fluctuation and reduces time complexity. On the other hand, a smaller basic unit increases time complexity, but reduces bit fluctuation. We propose an algorithm based on our study described in [117], where we determine the basic unit size adaptively, depending upon complexity of the scene.

The image quality of any video sequence is usually measured in terms of PSNR. It is found that PSNR increases with number of bits consumed in encoding. Thus, image quality improvement can be thought of as an optimization of bits and PSNR. Image quality is also measured in terms of

Table 2.8: Scratch Memory (Low) for Motion Estimation - II

Colour Component	Number of buffers	Total (bytes)
Y (9 x 9)	16	1296
U (3 x 3)	16	144
V (3 x 3)	16	144
	Total	1584

rate-distortion curve where PSNR is plotted against number of bits used in encoded stream. Rate distortion can be minimized by using rate distortion optimization (RDO) model, [118] which is very expensive computationally, and is not suitable for meeting real time the criterion on the selected DSP platform. In this section we shall first state all our assumptions, then discuss the state of the art of rate controlling, and finally define our proposed algorithm. Assumptions for implementation are as follows:

- To reduce complexity we consider each frame as a slice.
- Our approach can be applied to any H.264 encoder environment that is algorithms are not tailor made for target DSP.
- These algorithms are tested on different DSP platforms, and with different resolutions like QCIF, CIF, SDTV-525 test sequence.

Bit rate means the number of bits transmitted over network per second. In order to achieve a constant bit rate (CBR), we need to compute quantization parameter (qp), which depends on available bits and the mean absolute difference (MAD) between the original and the predicted image. Thus, bit rate (b) may be represented as:

$$b = f(qp, MAD) where 0 \leq qp \leq 51 \tag{2.1}$$

Rate control may be achieved only by manipulating qp, depending on MAD [119] and H.264 standards. [113] suggests that qp may vary in macroblock layer, slice layer or frame layer as delta qp is specified in all of these layer headers. So this rate controlling can be performed at a group of picture (GOP) level, frame level, slice level or macroblocks (MB) level. Each layer is treated as a basic unit for rate controlling. Theoretically, basic unit in rate controlling is defined to be a group of continuous MBs. However, MAD can be computed once reconstruction is done, but requires qp value. Thus, it is something like the chicken and egg problem. To come out of this dilemma, a linear model is used to predict the MAD of the remaining basic units in the current frame by using those of the co-located basic units in the previous frame. Suppose the predicted MAD of the l^{th} basic unit in the current frame and the actual MAD of the l^{th} basic unit in the previous frame are denoted by $MAD_{cb}(l)$ and $MAD_{pb}(l)$ respectively. The linear prediction model is then given by:

$$MAD_{cb}(l) = a_1 * MAD_{pb}(l) + a_2 \tag{2.2}$$

where a_1 and a_2 are coefficients. The initial value of a_1 and a_2 are set to 1 and 0, respectively. They are updated by a linear regression method similar to that used for the quadratic R-D model parameters estimation in MPEG-4 rate control after coding each basic unit. So now the problem of image quality enhancement is detecting the qp for a basic unit, depending on the predicted MAD and target bits, which are computed using the fluid traffic model which is clearly specified in the standard. It is also noted that by employing a big basic unit, a high PSNR can be achieved, though the bit fluctuation is also big. On the other hand, by using a small basic unit, the bit fluctuation is less severe, but there is a slight loss in PSNR. Besides, smaller basic units increase the computational cost. So it is better to perform rate controlling with frame as basic unit, when MB is not complex and with MB as basic unit when that frame contains complex picture information. Our contribution lies in selecting basic unit adaptively, depending on MAD, which optimizes bit distortion as well as time complexity, and thus, achieve higher PSNR. Here is a brief overview of how to implement our algorithm:

- Define qp as 28 for the starting frame.
- Compute average MAD at the nth frame ($MAD_{avg}(n)$) as $MAD_{avg}(n) = \frac{\sum MAD(i)}{n-1}$
- For every MB, compute the sum of absolute differences (SAD) between predicted and original pixel values and store it for future reference. As SAD computation is required in ME, no additional computational cost is required.
- Compute MAD for this macroblock using the equation in step 2. An MB is said to be a complex picture if it contains very detailed picture information.
- Complexity of an MB is quantitatively defined using a threshold-based approach.
- Threshold for n^{th} frame T(n) is defined as 80% per cent of ($MAD_{avg}(n)$). Threshold selection is done here using classical decision theory [120] based on 20 different test sequence of different resolutions.
- If the MAD for a particular MB is below the threshold it is said to be complex.
- The threshold is dynamically computed and it is updated at frame layer.
- If MAD $<$ T(n), that is, if the MB is complex, MB is chosen as the basic unit for that frame.

Table 2.9: Performance in Different Platforms

Processor	Clock speed	FPS QCIF	FPS SDTV
X86	2.8GHz	128	10
C55X	150Hz	12.4	1

- If MAD > T(n), for all MBs in the frame, basic unit is reset to frame layer.

2.7 Experimental Results

In this section the performance of different optimization and enhancement techniques are described and finally, the overall performance of the system is presented. Performance of the proposed method is not compared against [25], as the target DSP platform is different and thus, the performances cannot be compared. The hardware specifications of different testing environments are given in Table 2.9 and the memory utilization in those platforms are described in Table 2.10.

2.7.1 Algorithmic Optimizations

The graph in Figure 2.5 and Figure 2.6 show that our algorithm causes a significant reduction of MCPS at different bit rates and Figure 2.7 and Figure 2.8 show very insignificant loss in PSNR at different bitrates. These results show that we get significant MCPS reduction at the cost of negligible loss in image quality.

2.7.2 Use of Platform Specific Optimization

Optimization obtained using platform dependent optimizations are described in Table 2.11. It shows that the C level optimizations and use of EDMA has a significant impact to the process of optimizations. As external memory access is highly time consuming a huge optimization can be obtained if one can reduce the number of such accesses using EDMA.

Table 2.10: Memory Utilization

Processor	Code memory	Data memory	DARAM	SARAM
X86	-	-	128MB	-
C55X	62,107 Bytes	607,548 Bytes	64 Kbytes	256 Kbytes

Table 2.11: Performance of Platform Specific Optimizations

Platform dependent optimizations	Cycles per frame (QCIF) before applying optimizations	Cycles per frame (QCIF) after applying optimizations	Optimization in %
Use of Dual MAC	43410666.4	37176240.0	14.36
Use of intrinsics	35056886.1	31265534.3	10.81
C optimizations and EDMA	31265534.3	12468145.7	60.12

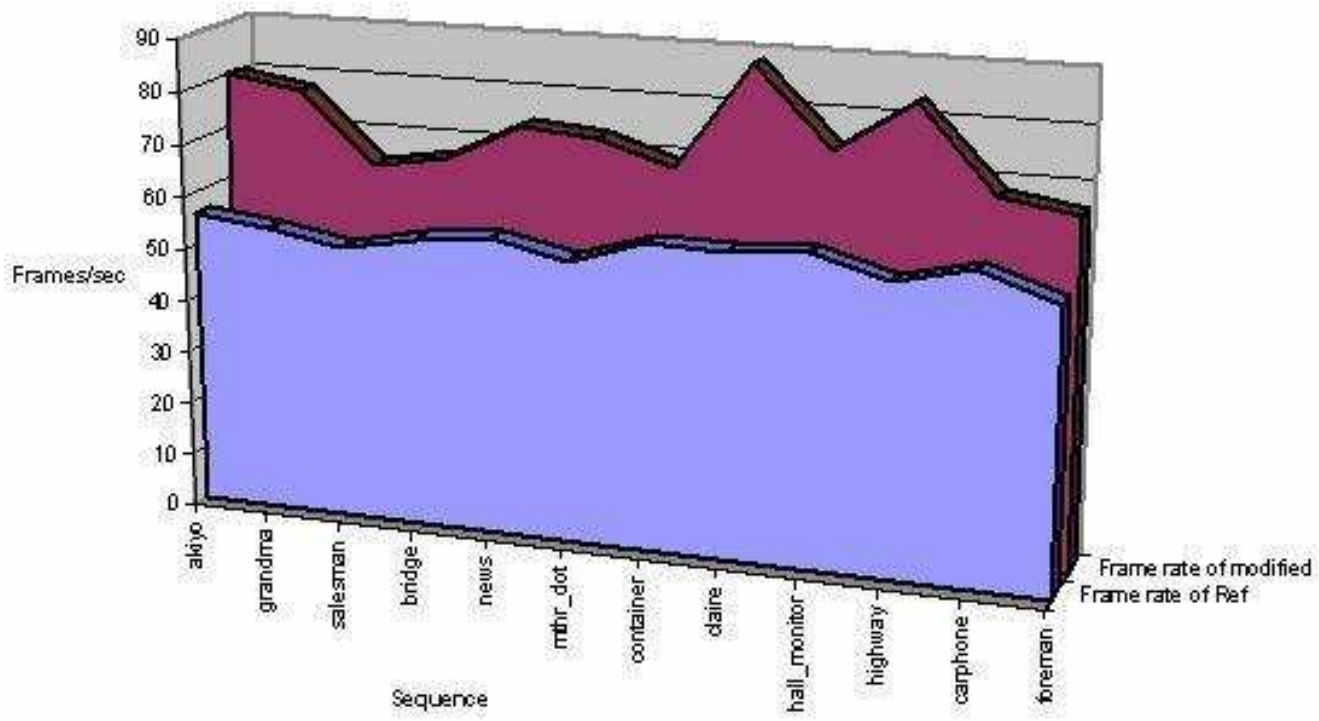

Figure 2.5: Sequence Wise Gain in MCPS for Videos Encoded at 32 kbps

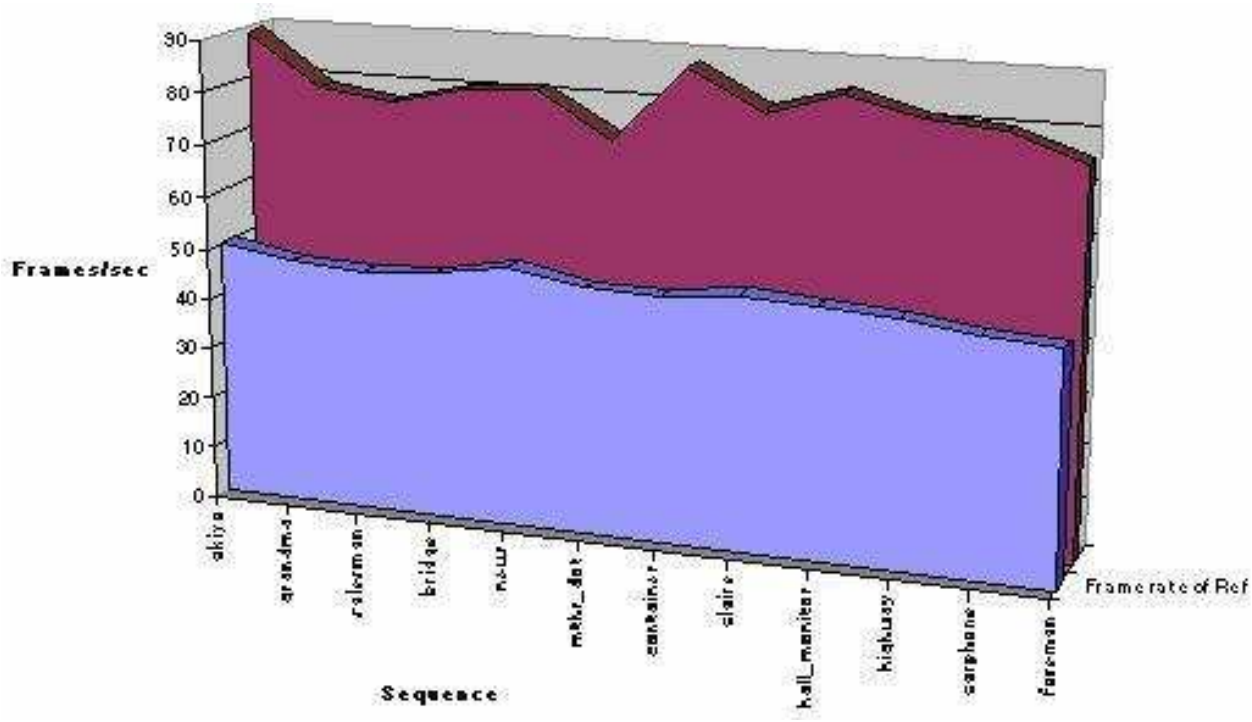

Figure 2.6: Sequence Wise Gain in MCPS for Videos Encoded at 128 kbps

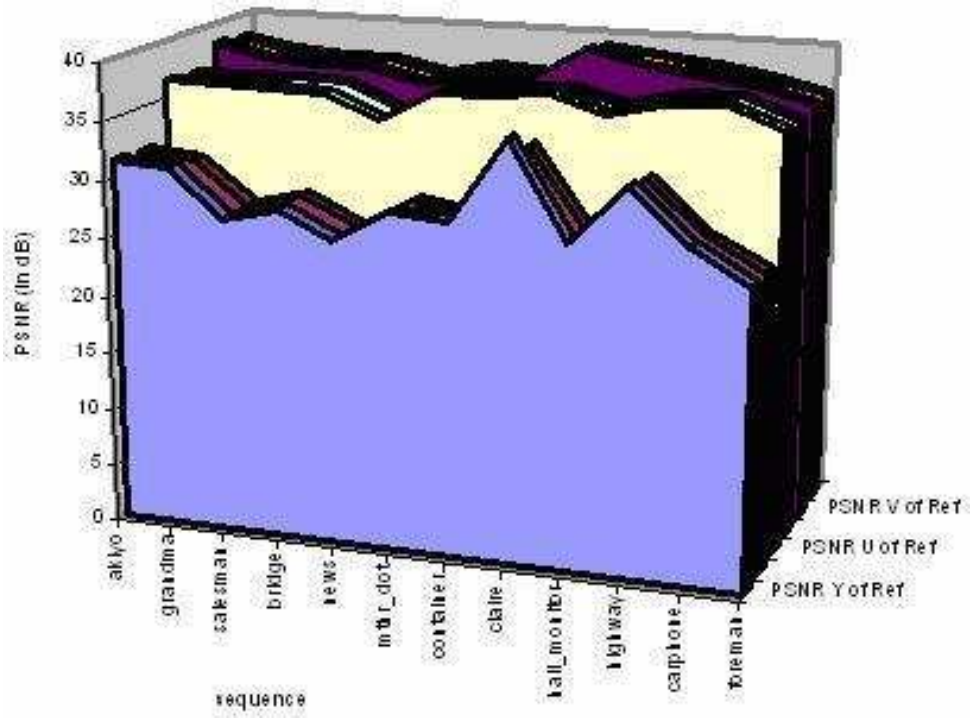

Figure 2.7: Sequence Wise Degradation in PSNR for Videos Encoded at 32 kbps

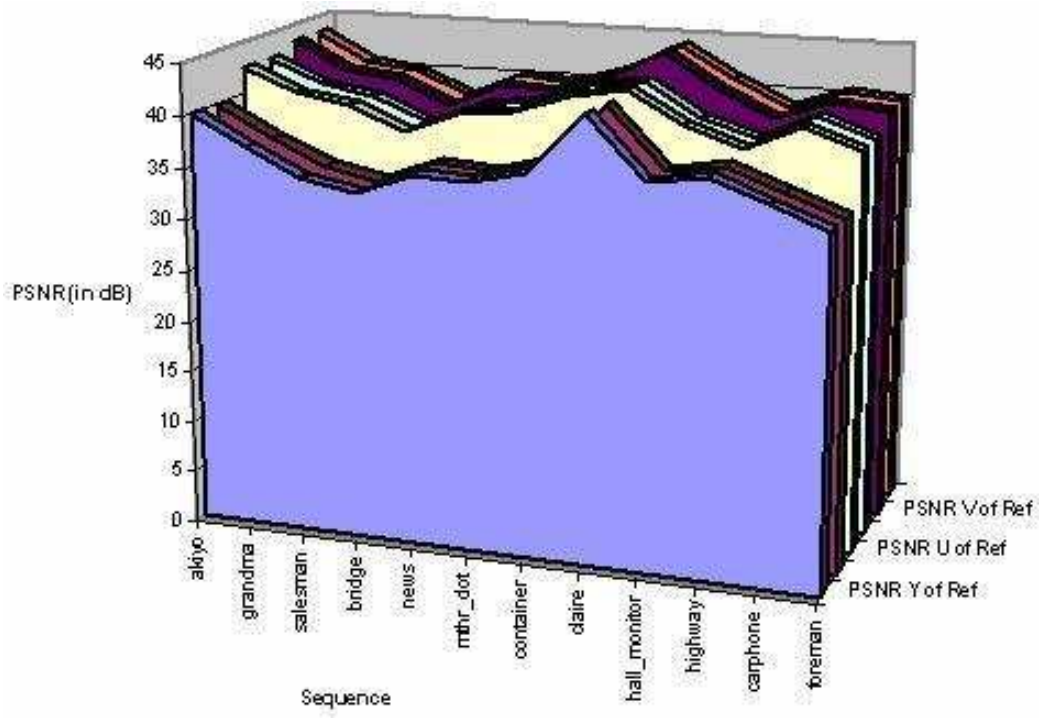

Figure 2.8: Sequence Wise Degradation in PSNR for Videos Encoded at 128 kbps

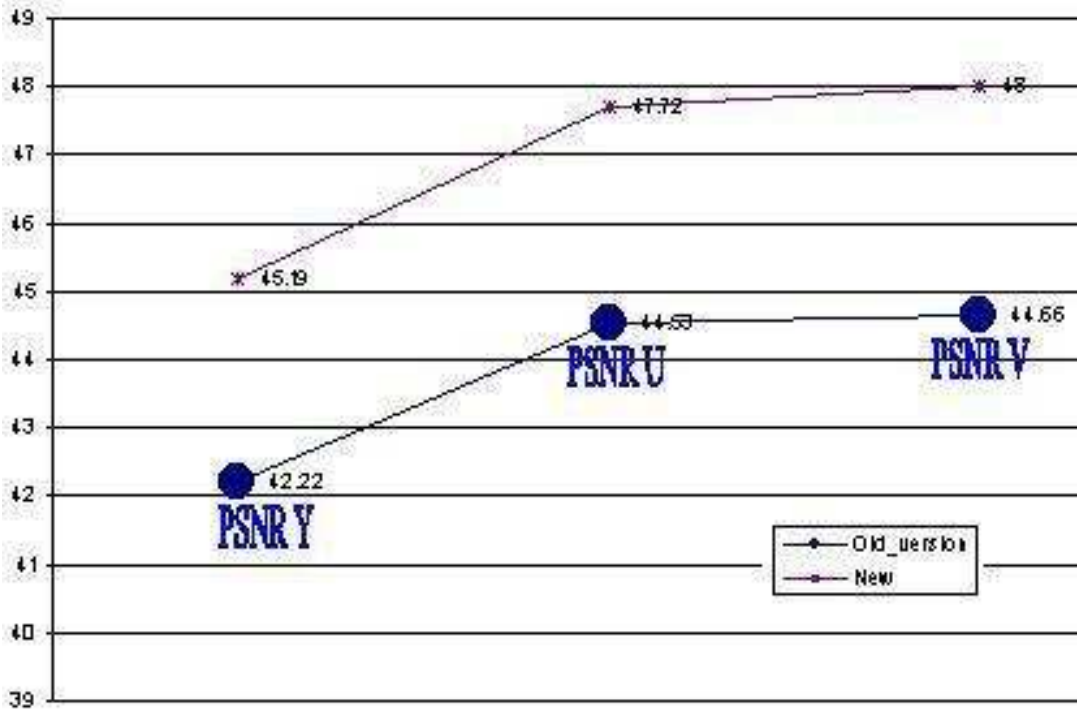

Figure 2.9: Improvement in Average PSNR

2.7.3 Use of Adaptive Basic Unit Selection for Rate Controlling

In Figure 2.9 the overall improvement in video quality using proposed adaptive rate controlling unit selection method is described. The performance comparison shows that the frame level rate control gives good results in some high PSNR areas and bad results in complex regions. Moreover, the bit can be handled by selecting MB as the basic unit, but this increases the time complexity. In our approach we select the basic unit adaptively and it gives less bit fluctuation and better performance in terms of speed and average PSNR, thus leading to an overall improvement in performance. Finally we show the snap shot of the developed video phone in Figure 2.10.

2.8 Some Use Cases Developed Using the Proposed H.264 CODEC

2.8.1 Video Conferencing

This video conferencing solution can be used by any enterprisers for inter organization communication. It can be also used for point to point video chat in the connected TV. One variant of the solution can be used in the Health care as a remote Video Consultation system having one-way high-resolution video and two-way audio, in addition to sharing medical sensor data from the patient side to the doctor at remote end. A similar system can also be used in Industrial Automation as a remote diagnostic system, with a high-end camera, which can support high-resolution video. The detail implementation of this solution is described in one of the related works of the thesis [121].

2.8.1.1 Features

The proposed solution has the following feature set:

- Server-less peer-to-peer architecture suited for enterprise networks
- RTP based streaming and A/V synchronization
- Support up to four simultaneous users using multicasting
- Collaborative document sharing and facility to store and replay complete sessions.

Figure 2.10: Snap Shot of Developed Video Phone

2.8.1.2 Platform Requirements

The following are the hardware and software platform requirements for the desktop video conferencing system.

- Hardware
  - Any DirectX compatible USB webcam
  - PC Speaker or Headphone and Microphone
  - DaVinchi Processor
- Software
  - Microsoft Windows XP with Direct X 9.0 version C or higher
  - DirectX drivers for web cam, audio recording and audio playback)

2.8.2 IP Video Phone

One of the major applications for the advanced video compression technologies is the Internet Protocol based stand-alone Video Phone. With a low-cost implementation of this, it is feasible to provide video telephony facility in every household and commercial enterpriser, thereby reducing the distances between people situated across different geographies. Low cost solution with low bandwidth consumption is the key requirement.

2.8.2.1 Features

- Cost-effective and low-power video telephony solution
- H.264 based video compression and AMR-NB based speech compression
- TCP/IP based communication
- Echo Control for Voice

2.8.2.2 Platform Requirements

- TI's TMS320c55x as the DSP core TI OMAP 1610
- ARM9 as the GPP core

- USB connectivity with Camera
- On-board audio capture
- On board playback
- Colored LCD screen for display
- Linux (kernel version 2.4) (O.S)

2.8.3 Place-shifting System

Central idea of the Place-shifting system is to provide the user with an access to his home video content viz., cable TV, DVDs, VCDs etc. on his laptop over broadband even when the user is geographically far away from home.

2.8.3.1 Features

- Capture of analog composite video/audio from TV/Cable Set top Box/PVR Box
- Encoding of captured video stream using H.264 and captured audio stream using AMR-WB
- Control of QoS/Bandwidth Requirement through constant bit rate encoding
- Streaming over broadband to remote PC/Laptop using TCP/IP
- Audio/Video player application on PC/Laptop with display of the video up to full-screen
- Possibility of controlling TV/Cable Set top Box/PVR Box by emulating the remote controller as laptop taking help of IR Blaster from remote end.

2.8.3.2 Platform Requirements

- TI DM642
- TI c6x based DSP
- Suitable Flash
- SDRAM
- Video Capture Port with DMA

Table 2.12: Performance of Desktop Video Conference

Video quality (QCIF)	36.16 dB
Resolution	QCIF (176x144) to SDTV625 (720x576)
Bandwidth	37 kbps for one-way audio/video for QCIF
Storage	0.25 MB/minute of video
VBR/CBR	Both support
Error resilience	Robust against packet loses
Encryption	Supported
Watermarking	Supported
Leap Synchronization	Supported
Speech CODEC	AMR - NB

- Audio Capture Port with DMA
- Video Encoder (for A/D conversion)
- Audio A/D converter
- Ethernet port
- GPIO port for controlling IR Blaster
- PC/Laptop
- Microsoft Windows XP
- Direct X 9.0 version C or higher

2.8.4 Performance

The performance of the proposed systems are summarized in tabular format in Table 2.12, Table 2.13, Table 2.14.

Table 2.13: Performance of Video Phone

Video quality	40.61 dB (For Y comp and QCIF resolution)
Resolution	QCIF (176x144)
Power	0.2 Watt
Frame Rate	5
Bandwidth	37
Internal memory requirement	DSP: DRAM 64KB, SARAM: 96KB
VBR/CBR	VBR
Error resilience	Robust against packet loses
Speech CODEC	AMR - NB
Voice Activity Detection	Supported

Table 2.14: Performance of Place Shifting

Image quality	40.61 dB
Resolution	CIF (352x288)
Power	2.15 volts @ 60% CPU utilization
Frame Rate	15
Bandwidth	256 kbps
VBR/CBR	Both supported
Error resilience	Robust against packet loses
Leap Synchronization	Supported
Speech CODEC	AMR - NB

Chapter 3

Video Security for H.264

3.1 Introduction

In this chapter some more value added services for STB have been proposed. One of the major features of recent STB is that they can store the video content into some harddrive. This storage of STB is commonly known as Personal Video Recorder (PVR). But when the TV shows are recorded into a storage device, the legal issue comes with the copyright of those broadcast TV contents. It is quite likely that any customer of such a system can illegally sell the contents after recording it into his/her personal device which is a threat for the content providers. So some suitable method to prevent the copyright should be implemented along with the video encoder so that no Intellectual Property Right related issues can arise. One way to do this is to apply some encryption and watermarking techniques to ensure the security of the multimedia content streamed by the channels. Encryption allows the user to get the video only with valid keys while the watermark would ensure the copy right prevention. Moreover this security should be implemented in real time by inserting the watermark and encrypting the video during storing the video into PVR and thus there is no additional overhead is there. Over and above the security should be imposed without any significant increase in bandwidth. So our second chapter is dedicated to implementation of security measures in real time on the same target DSP platform and compatible to H.264 video encoder. Moreover we shall propose the method of evaluating a video watermark which was also developed as a part of this study.

Several studies on encryption and watermarking is already described in Section 1.1.2. So, in this chapter initially a video encryption technique for H.264 video is proposed and then a watermarking technique is proposed. The method for evaluating any video watermarking technique is also described in this chapter. Proposed video encryption algorithm can encrypt any video in real-

time while encoding it into in H.264 format. This encryption algorithm is computationally efficient and so that it can be implemented on any commercially available DSP platform. The method of watermarking involves a blind watermarking scheme that inserts watermark in transformed coefficients of Independent Decoder Refresh (IDR) frames of H.264. As the IDR frames are required to reconstruct the P frames, the proposed scheme can ensure security for both the frame types (i.e. I and P) of the baseline profile of H.264. The method also uses a hashing technique to hash the entire group of pictures (GOP) into a 128-bit number. This number, in turn, is used as watermark for the next GOP to ensure the integrity feature of the watermarking scheme. Subsequently a method to evaluate a video watermarking technique by evaluating the video quality after watermarking and also the robustness of the video watermarking scheme is discussed in this chapter. In the method of evaluating a watermark, initially the original video is compared against the watermarked video stream using different factors and finally a single decision is taken out of those multiple factors. Similarly the robustness is evaluated using multiple factors. Usually any binary image or any text message is used as the watermark. In the proposed approach binary image and/or text message (used as watermark) is compared with the binary image and/or text message retrieved from the attacked video and assign some matching score to it. These scores are then compared with the results obtained by Mean Opinion Score (MOS), which is purely based on Human Vision Psychology (HVS). Finally an adjective factor membership value is assigned to the watermarking technique under review based on these two features.

3.2 Video Encryption Technique Applicable to H.264 AVC

In the proposed method, encryption of H.264 is achieved while encoding a video into H.264 format. In this method encryption is achieved in two folds, namely the header and the video content using the architecture of H.264 AVC.

3.2.1 Header Encryption

Basic processing unit of compressed H.264 video data in decoder side is Network Abstraction Layer Unit (NALU). Structure of a NALU is defined in 3.1. Any NALU can contain either video data or control data. The syntax of the read module is shown in Figure 3.1. Control information like profile, label, height, and width are written in the header part. The formation of NALU depends on application and network layer, also. For example if there is a high chance of packet loss,

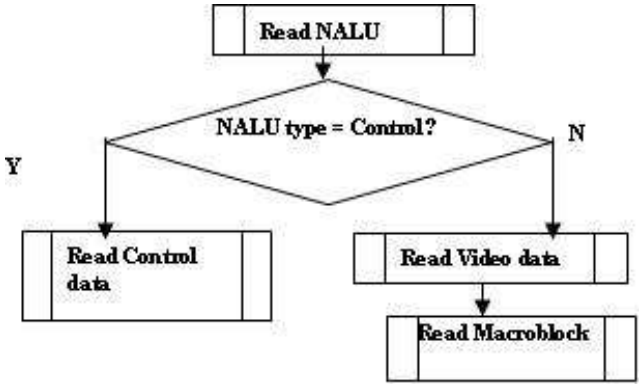

Figure 3.1: Block Diagram of Read Module

control data (like Sequence Parameter Set (SPS), Picture Parameter Set (PPS)) and Independent Decoder Refresh (IDR) frames are transmitted repeatedly so that if one header gets corrupted it can recover as soon as the next set of SPS and PPS arrives. On the other hand in video conferencing like applications, where one user can join while the conference between two has already started, synchronization requires the presence of multiple SPS, PPS. Again in case of video storage, fast forward and rewind utility requires the presence of IDR at every 1 or 2 seconds. Some typical NALU organizations may be like Figure 3.2 or like Figure 3.3.

Now, the NALU type can be any of the following shown in 3.1. as suggested by H.264 standard [113]. Among the different types of NALU, SPS and PPS contains the control information without which it is not at all possible to start decoding. SPS contains information like profile, level, picture order count type etc and PPS contains information like height, width etc. So unless decoder gets this information it cannot start the decoding process. Our encryption method encrypts these SPS, PPS and IDR as described below:

- Take a 16-bit user code (K_U) that will be shared through secured medium like e-mail or phone. This user code can be obtained from the hardware identification number of the HIP, too
- Encode the first frame using conventional H.264 encoder

Figure 3.2: NALU Organization for Application like Video Conferencing

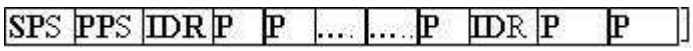

Figure 3.3: NALU Organization for Application like Video Storage

- Take the length of IDR(l_{IDR}) (It is a 16-bit number for QCIF resolution (176x144)) as the private key
- Define Key value K_P using a Hash function (H) of l_{IDR} and K_U
- $K_P = H(K_U, l_{IDR})$
- Mask SPS, PPS, IDR with this key value i.e. K_P

3.2.2 Video Content Encryption

The overall process of H.264 encoder is described in Figure 3.4. Proposed encryption algorithm for picture level plays at FMO. For the simplicity of design and time complexity of the overall system, it is considered that there is no slice partition in a frame i.e. a slice is same as a frame. So in the existing H.264 Encoder system (without encryption), only one map unit type 0 is used. Now the detail of conventional FMO is described using Figure 3.5 (left). In the modified design for implementing encryption into H.264 encoder the block diagram becomes as Figure 3.5(right).

The proposed encryption algorithm is based on some assumptions as listed below:

- H.264 video encoder/decoder to be used are of baseline profile
- For simplicity of design, there is no slice partitioning in a frame. However the algorithm works fine even in the presence of multiple slices.
- Any Group of picture (GOP) consists of 15 frames or 30 frames.
- Encryption algorithm is applicable to Independent Decoder Refresh (IDR) frames only.

The proposed encryption algorithm works in the FMO block of H.264 video codec [113] to get next MB number. The pseudo code for the encryption algorithm is as below:

- Take a 16-bit number from user as key (K_P)
- Store the length of the encoded bitstream for the last Independent Decoder Refresh frame (IDR) as l_{IDR} which is a 16-bit number for QCIF resolution (176x144).

Table 3.1: NALU Unit type

NALU type	Content of NAL unit and RBSP syntax structure
0	Unspecified
1	Coded slice of a non-IDR picture slice-layer-without-partitioning-rbsp( )
2	Coded slice data partition A slice-data-partition-a-layer-rbsp( )
3	Coded slice data partition B slice-data-partition-B-layer-rbsp( )
4	Coded slice data partition C slice-data-partition-c-layer-rbsp( )
5	Coded slice of an IDR picture slice-layer-without-partitioning-rbsp( )
6	Supplemental enhancement information (SEI) sei-rbsp( )
7	Sequence parameter set seq-parameter-set-rbsp( )
8	Picture parameter set pic-parameter-set-rbsp( )
9	Access unit delimiter access-unit-delimiter-rbsp( )
10	End of sequence end-of-seq-rbsp( )
11	End of stream end-of-stream-rbsp( )
12	Filler data filler-data-rbsp( )
13 .. 23	Reserved
24 .. 31	Unspecified

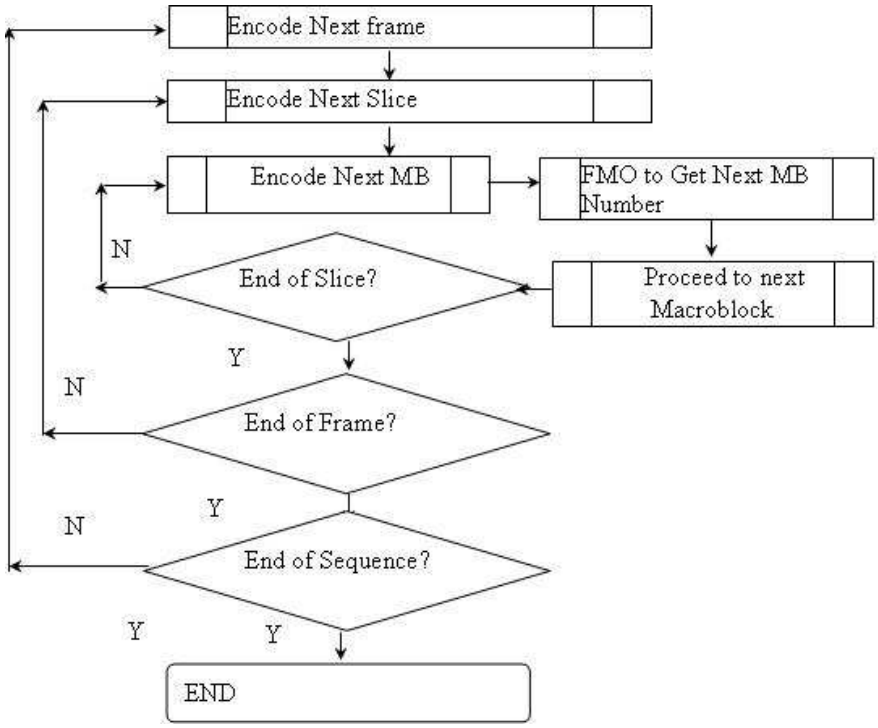

Figure 3.4: Flow Chart of Process of Encoding

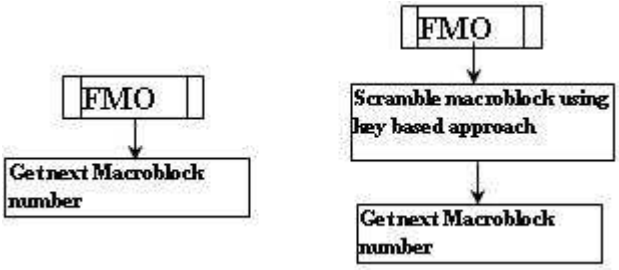

Figure 3.5: Original FMO Algorithm and Modified FMO Algorithm

- Obtain a new value for using a Hash function (H_f) of l_{IDR} and K_P for the next Group of Picture (GOP).
- $K_P = H_f(K_P, l_{IDR})$
- Use K_P to generate an ordered random sequence $L_e = 0, 1, 2, ..., 97$ using the any random number generation algorithm. We keep the last MB no as 98 as we support single slice and for single slice last macro block number should be 98.
- In encoder side, MBs in an IDR frame are encoded in the order specified by a look-up table which is a random permutation of the sequence 0, 1, 2, .98.
- In encoder side the MB to be encoded (current MB) is predicted from image samples that have already been encoded. The decoder creates an identical prediction and adds this to the decoded residual or block. So if the order of decoding the MBs is not same with the order of encoding the MBs, decoder may try to predict an MB using an MB that is not available at that time and thus the decoding process halts.

3.2.3 Security Analysis

In the proposed algorithm, MBs of a video frame are encoded in the order specified by a look-up table which is a random permutation of the sequence 0, 1, 2, .98. But the decoder can decode the frames only if the look-up table at the decoder side (L_d) is identical to L_e. While decoding a particular MB, the decoder may not get the MB information using which the current MB was predicted if $L_e \neq L_d$. Now, as per the theory of permutation, L_e can be constructed in 98! ($= 9.42 \times 10^{153}$) ways. So if the hacker wants to generate the decoder look-up table which will be equivalent to L_d, he needs to make 98! attempts to successfully decode the video. But this

Table 3.2: Computational Complexity

Operation	Frequency per GOP	% Increase with respect to H.264/AVC
ADD	2*24*97 + 3 = 4659	0.004
MULTIPLICATION	5*24*97 = 11,640	0.200
DIVISION	5	0
MODULO	4*24*97 + 4 = 9316	0

theoretical complexity is restricted by length of the user key. In that implementation, we have used a 32 bit number as the key. So L_e can be generated in 4294967296 ways. So, on average 2147483648 numbers of tries is required to decode the first GOP correctly. But this key will be changed after one GOP. So it can be concluded that the encryption method is robust enough in comparison to the previous work described in [32] where the authors have claimed that a hacker can break the security in 2128 tries.

3.2.4 Complexity Analysis

3.2.4.1 Computational Complexity

Time complexity of the encryption algorithm is same as that of the random number generation algorithm. We shall discuss this in terms of basic operations like ADD, MULTIPLICATION, DIVISION, MODULO. We assume that memory input-output operations and control operations like if -then-else are less complex than the above mentioned operations. We shall discuss the complexity in the [32]. The initialization module, which is used for seed computation takes this much amount of time complexity. The results of our proposed algorithm are presented in Table 3.2. There is no such figure available that can be used to benchmark our algorithm.

The modulo operations can be reduced with the help of look-up tables. We have tested the system in PIV 2.8 GHz processor and seen that the H.264 encoder with and without encryption can ran at (on average of 5 runs) 145 Frames per second. So it also proves that there is no significant computational overhead for the proposed method. But there is no such figure available for the previous work on H.264 based video encryption technique depicted in [32]. So it clearly gives an improvement in time complexity in comparison to the over head of .188 to .309 millisecond as reported in [32].

Table 3.3: Space Complexity

Resolution (wxh)	Picture size in MBs (wxh/(16x16))	Memory requirement (in bytes)
QCIF (176x144)	99	198
CIF (352x288)	396	792
VGA (640x480)	1200	2400
SDTV-525 (720x480)	1350	2700
SDTV-625 (720x576)	1620	3240

3.2.4.2 Space Complexity

An additional storage for look up table is required. That table consists of the macroblock number. For scalability to CIF (352x288) and higher resolution the type of that array should be 16bit storage. The size of the array should be equal to the size of that resolution in macroblocks. So if the resolution be wxh then the additional storage requirement is wxh*2/(16x16) bytes. The results of our proposed algorithm are presented in Table 3.3.

3.2.4.3 Compression Vs Video Quality Performance

Considering the process of encryption, our algorithm has almost no effect on compression ratio and also the video quality expressed in terms of PSNR. The performance of our proposed method is described in Table 3.4. We can also prove this by comparing the size of un encrypted files and encrypted ones, as shown in Table 3.4. The sequences were encoded by the video encoder (VBR version)described in chapter 2. The selected parameters are as: Quantization Parameter (QP) = 28, interval of IDR frame = 30. We take the average of the result obtained by running the video sequence on 200 frames.

Our algorithm shows that there is an increase of nearly 1% in bit rate if we use our encryption algorithm by keeping the same video quality. There is no such figure available for the existing works.

Table 3.4: Comparative Analysis of Encrypted and Un-encrypted Algorithm

Video Sequence (90 frames)	Size Without encryption	Size With encryption	PSNR Without encryption	PSNR With encryption
Claire	155.125 kb	158.16 kb	39.03 db	39.03 db
Foreman	668.975 kb	700.3 kb	35.14 db	35.16 db
Hall monitor	264.82 kb	268.705 kb	36.97 db	36.96 db

Yuan Li et al [32] shows that there is nearly 1% increase in bitrate for their algorithm but they did not give any figure for their video quality.

3.3 Video Watermarking technique Applicable to H.264 AVC

3.3.1 Proposed Watermarking Algorithm

Basic description of the watermark module and its interaction with H.264 CODEC is described using block diagram. In Figure 3.6. Here the gray colored blocks are the addition for embedding watermark. In the proposed watermarking method, watermarking in H.264 encoder is achieved in two folds: in even numbered Independent Decoder Refresh (IDR) frames, three different types of watermark messages are embedded and in odd numbered frames the bitstream is obtained by hashing the last Group of Picture (GOP) to ensure integrity. It is also depicted in pictorially in Figure 3.7. The process is described in details as follows:

- Is the frame an even numbered IDR?
- If Yes, embed logo, timestamp and IP address or key.
- Else Hash the last GOP and embed the number.

The various sub-blocks of the algorithm in Figure 3.7 are described in detail below.

3.3.1.1 Test for Watermarking Message Size

This process of checking the size is depicted in Figure 3.8.

3.3.1.2 Find the Location for Embedding Watermark

Selection of the position of DCT coefficient to be modified is based on our following observations:

- Most significant information lies in top and left
- Modification of diagonal elements at right and bottom results in insignificant artifacts
- Coefficients in diagonal positions are more stable than the others.
- Embed one watermarking bit in one diagonal coefficient of diagonal sub-block

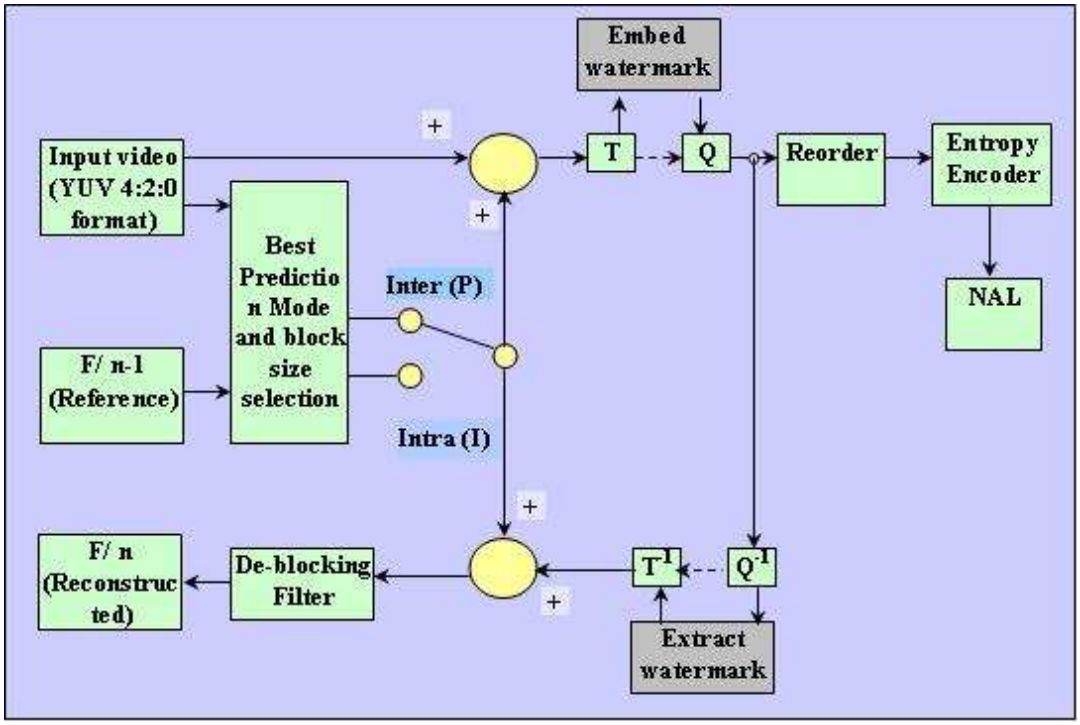

Figure 3.6: Overview of Watermarking Method

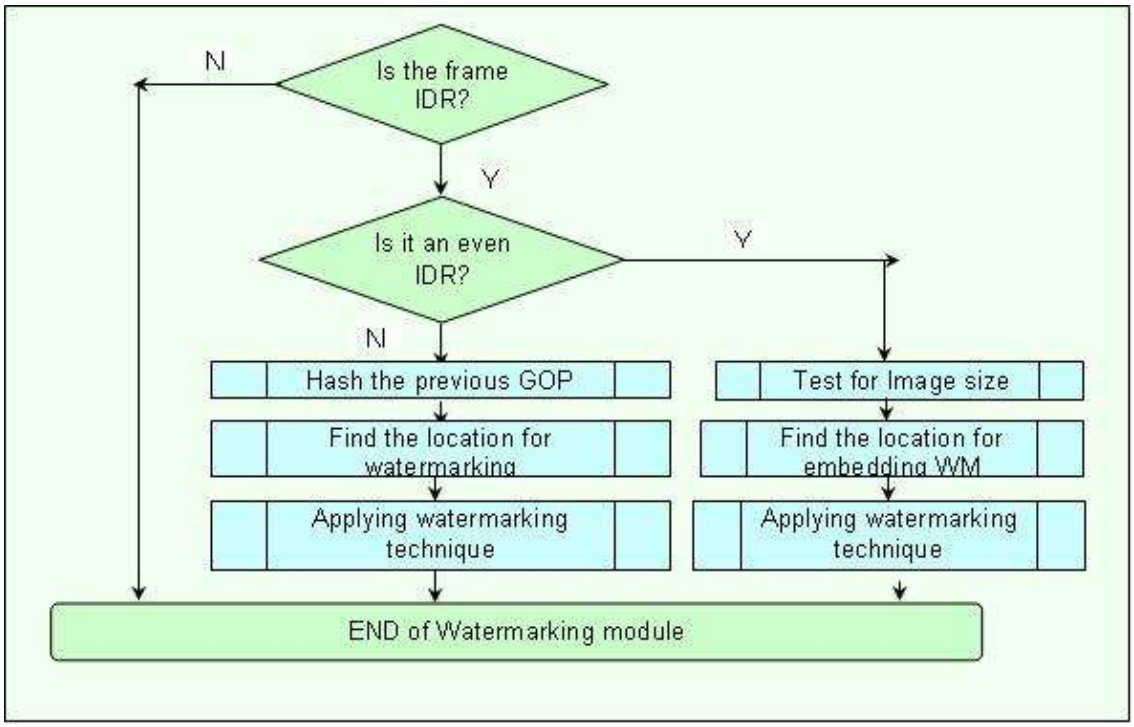

Figure 3.7: Details of Watermarking Method

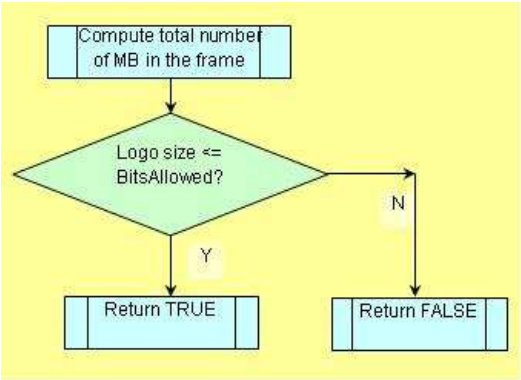

Figure 3.8: Test for Watermarking Message Size

3.3.1.3 For Inserting Binary Image Like Logo

As the binary image has a higher payload compared to the text information, diagonal sub blocks are used.

- Find whether the sub-block is diagonal
- Every sub-block has 16 coefficients (4x4)
- Embed the watermark in 10th or 15th coefficients only
- If the bit number to be embedded is odd insert it in 10th coefficient
- Else in 15th Coefficient

3.3.1.4 For Inserting Text Message Like Time Stamp and IP Address

- Find whether the sub-block is ab-diagonal
- Embed the watermark in 10th or 15th coefficients only
- If the bit number to be embedded is odd add it in 10th coefficient
- Else in 15th Coefficient

This process is described in Figure 3.9 and Figure 3.10.

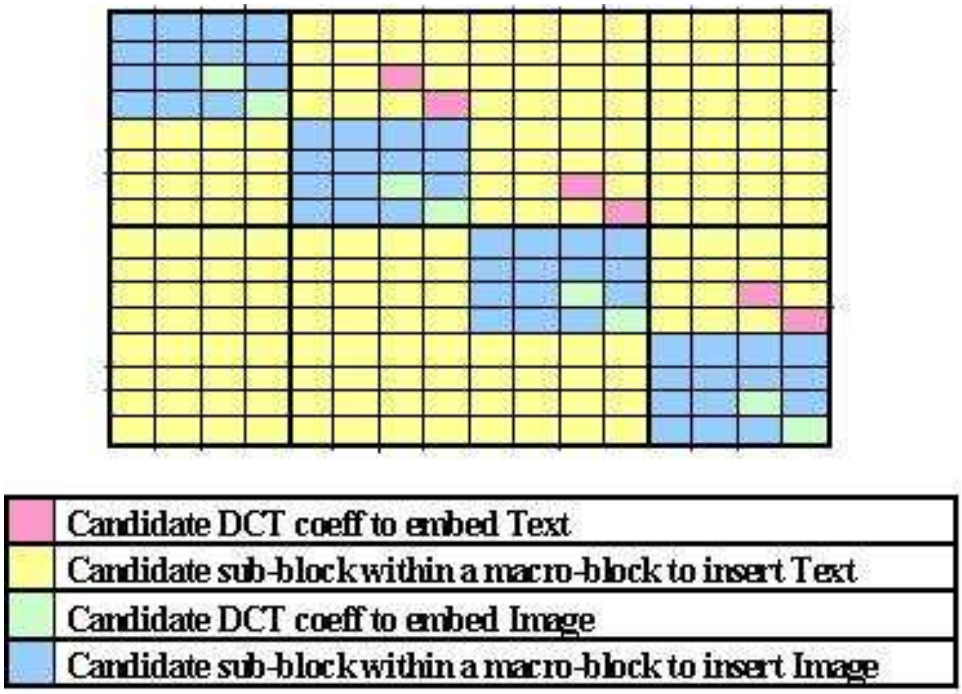

Figure 3.9: Finding Suitable Candidate for Inserting Watermark

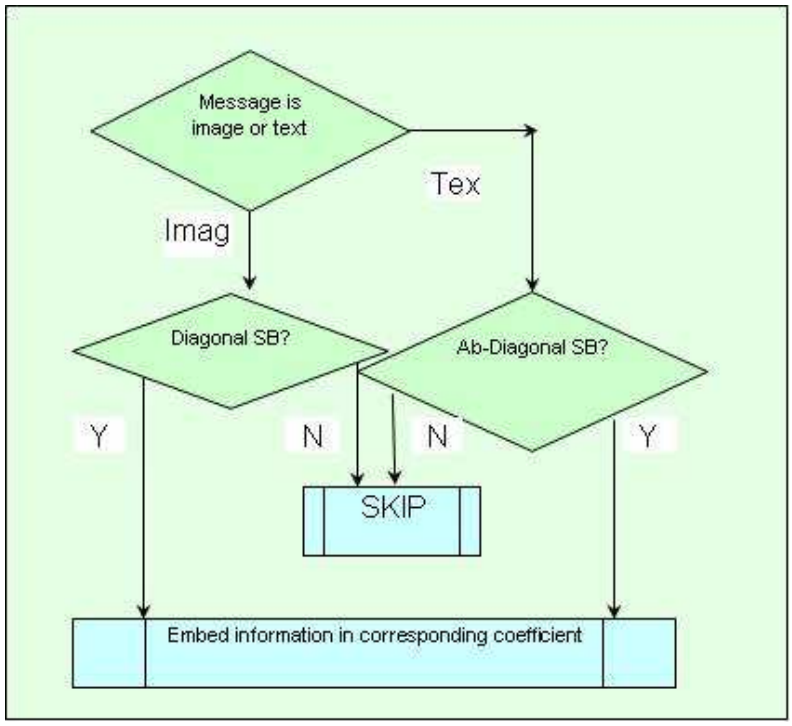

Figure 3.10: Finding Suitable SB for Inserting Image/text

3.3.1.5 Watermark Embedding Technique

We have used two types of information that needs to be inserted as watermark into the video. They are (i) binary image (like logo), (ii) Text information (like time stamp, IP address). The logo (image) of size 24x16 used as Watermark embedding information is stored in binary format, i.e. it contains only 0 (for black) or 1(for white). We have used I/P address and time stamp as text information. As each of these two namely IP address and time stamp are of size 32 bits, the text information embedded as watermark is of length 64 bit. So the payload size is 24*16 + 64 = 384 + 64 = 448 bits for a QCIF (176x144) video. Here is the method for watermark insertion:

- Store the watermark information in a 448 byte array (we call it w_n) whose each byte is either 0 or 1.
- Quantize w_n using same quantization parameter (QP) as used in the video.
- Store the quantized values in another array (w_{qn} of size 448)
- For each w_{qn} , find the location of embedding (already discussed in earlier section)
- If w_n is 1 Find MAX(quantized video coefficient, w_{qn}) and replace the video coefficient by Max value else Make the video coefficient 0.

Two basic intra-prediction modes INTRA-4x4 and INTRA-16x16 are used in H.264/AVC, comprising 4x4 and 16x16 block-wise directional spatial predictions, in which the pixel value of the current block is predicted by the edge pixels of the adjacent blocks. Then, the prediction error is transformed primarily by a new 4x4 integer DCT instead of the float 8x8 DCT, which is widely used in previous video compression standards like MPEG. The smaller block-size is justified by the advance of prediction capabilities by using above mentioned prediction modes, but it makes the watermarking scheme more sensitive to attacks. Our proposed watermarking algorithm is as follows:

- Insert watermark bits by altering the quantized AC coefficients of luminance blocks within I-frames. In order to survive the re-compression, two major obstacles are considered as follows.
- First of all, the watermark signal $M(u, v)$ must be strong enough to survive the quantization, so that

$$|M_q(u, v)| = |quant[M(u, v), QP]| \tag{3.1}$$

Table 3.5: Complexity Analysis of Watermarking Algorithm

Operation	Number of operations per GOP (1 GOP = 30 frames)
Add	2779
Multiplication	3564
Division	1980
Modulo	3564
Condition(like if-then-else)	7524
Memory read/write	1584

- Where the quant[.] denotes the quantization operation, QP denotes the quantization parameter and (u, v) denotes a position in a 4x4 block B_K) Obviously $M(u, v)$ should be even greater if the watermark is required to survive the requantization during transcoding.

- Furthermore, since the change of the prediction direction (or mode) during transcoding may alter the value of DCT coefficients and thus leads to watermark detection error, we choose one of the quantized AC coefficients $X_q(u, v)$ in high frequency along the diagonal positions (i.e., u = v) for embedding. Our experiments show that the coefficients in diagonal positions are stabler than the others. Thus, the $X_q(u, v)$ is replaced by the watermarked coefficient X.q ,

$$X_q = max(X_q(u, v), M_q(u, v)) \quad if \quad w_n = 1; 0 \quad if \quad w_n = 0 \tag{3.2}$$

- where w_n is the bit to be embedded. We notice the AC coefficient $X_q(u, v))$ is cleared if '0' is embedded. It can be justified by the fact that the $X_q(u, v)$ is zero in most cases. It will not introduce significant artifacts.

- After watermarking, the best mode for a watermarked macroblock is selected by minimizing the modified Lagrange optimization function:

$$J_{Mode} = D_{REC}(S_k^*, I_k) + \lambda_{Mode} R_{REC}(S_k^*, I_k) \tag{3.3}$$

- where D_{REC} and R_{REC} represent the distortion and the number of bits, respectively, encoded for modes $I_k \in \{INTRA - 4 \times 4, INTRA16 \times 16\}$. λ_{Mode} is the Lagrange parameter

Table 3.6: Implementation Results (QCIF, H.264 Baseline Profile):Watermark Embedding

Platform	Configuration	Application	Mega Cycles per GOP
Pentium 4	P4 2.8 GHz	Desktop Video Conferencing	14.0
64X	DM642, 720 MHz GHz	Place shifting (15 FPS)	8.028

3.3.2 Evaluation of the Proposed Method

Theoretical complexity of the algoritm is decsribed in terms of some basic operations like Add, Multiplication in 3.2. Complexity in terms of Mega cycles per Group of Picture (GOP) in different platforms is shown in 3.6. The video quality and robustness is described in later part of this chapter.

3.4 Evaluation of Quality of Video Watermark

Now we are going to discuss the method of evaluating any video watermarking technique. Details of the method are described in subsequent sections.

3.4.1 Architecture of Video Watermark Evaluation System

Watermarking can be embedded on a video in either compressed domain or in pixel domain. On the other hand the watermarked video can also be in either raw format or compressed format. The proposed video watermarking evaluation method can work for both the cases. The steps involved in evaluating the robustness of a video watermarking scheme are as below:

- Check whether the input video is in compressed format or in a raw format.
- If the input is in compressed format, then we initially decode it to raw format.
- Apply attacks on the input watermarked video so that the attacked video also comes in raw format.

Table 3.7: Implementation Results (QCIF, H.264 Baseline Profile):Watermark Extraction

Platform	Configuration	Application	Mega Cycles per GOP
Pentium 4	P4 2.8 GHz	Desktop Video Conferencing	14.0
64X	DM642, 720 MHz GHz	Place shifting (15 FPS)	8.028

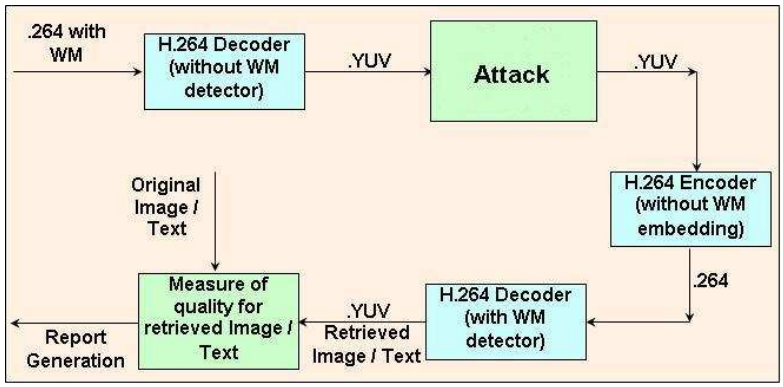

Figure 3.11: Block Diagram of the System Architecture

- If the watermark extracteor runs on compressed video then we first compress the attacked raw video into the required format
- Apply the watermark extractor module on the attacked video
- Generate the report on the video quality after attack by comparing the attacked video and the watermarked video
- Generate report on robustness by comparing the retrieved watermark and the original watermark
- Conclude on the robustness of the watermarking scheme under test

The architecture of the tool is given in Figure 3.11.

The method proposed in [122] is generic enough to support any video watermarking system. In the proposed system we have implemented the modules as described below:

- Some benchmarking attacks are used to simulate the attack module
- Reference H.264 Encoder and Decoder are taken from JM [123]
- H.264 decoder with Watermark extractor developed by us as described in earlier section of this chapter.

The attack module is simulated as described in Figure 3.12. Here we have used some benchmarking image watermarking attacks implemented in Stirmark. But the Stirmark works on the input images in .BMP format only. So the following steps are followed:

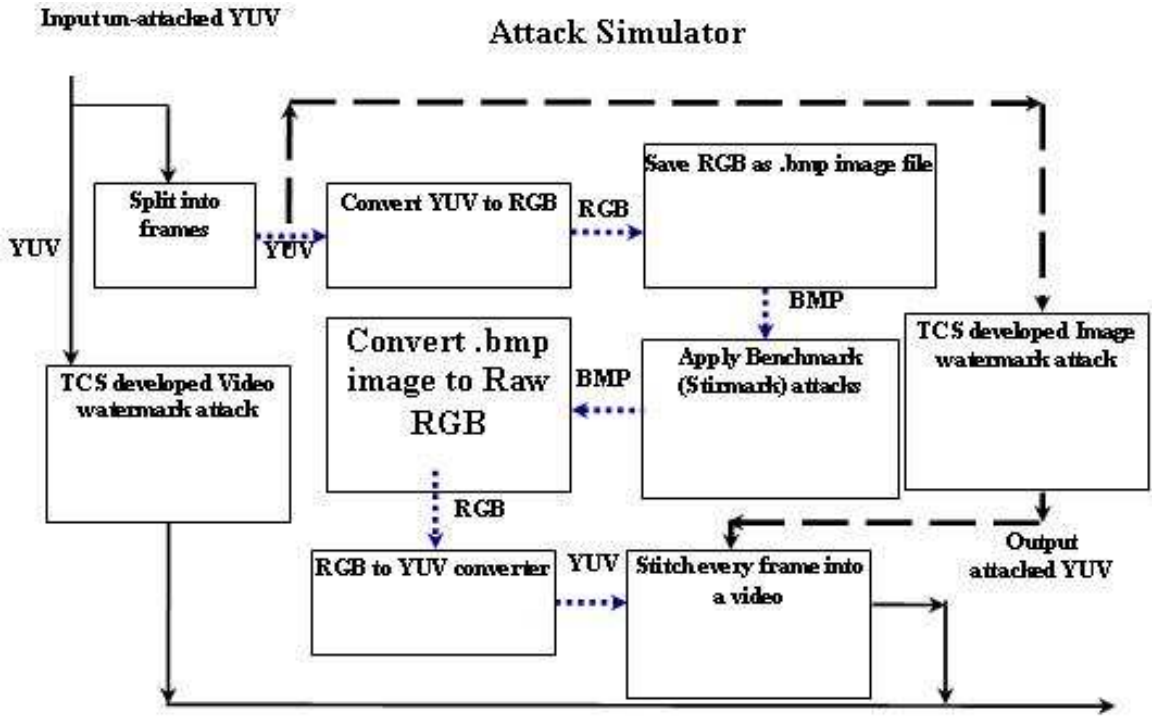

Figure 3.12: Block Diagram of Attack Simulator

- Split the input video into frames
- Convert the color space of each frame from .YUV to RGB
- Save each frame as a BMP image
- Apply Stirmark attacks and some in-house developed attacks

As any video sequence can be thinking of as a series of still images, the attacks applicable to images [122], [77]-[84], [65], [72]-[76], [85] can be applied to videos also. We have used the following attacks along with the Stirmark attack to simulate the attack module.

- Averaging attack (AA), Circular averaging attack (CAA), Rotate attack (RoA), Resize attack (RsA), Frequency filtering attack (FFA), Non linear filtering attack (NLFA), Gaussian attack (GA), Gama correction attack (GCA), Histogram equalization attack (HEA), Laplacian attack (LEA)
- Convert the color space of output attacked images from RGB to YUV
- Stitch the frames to form a attacked video

3.4.2 Evaluation of Video Quality Factor

The list of acronyms used in rest of the discussion is given in Table 3.8. In this section we shall discuss the method how the video quality is evaluated by comparing the original video with the watermarked video. Different image comparing metric can be found from [63], [85] but they didn't focus on video. Moreover these features are just depicting some values and some of the features are working for some cases only. So we have proposed an approach where we have used ten features to conclude on the video quality. We have used multifactorial approach to compare the video quality. Here we consider each frame as an object and each object is evaluated against ten factors as described below:

3.4.2.1 Description of the Selected Features

The list of features, their description is described in the following sub-sections.

Table 3.8: List of Acronyms Used

Acronym	Full name
AAD	Average Absolute Difference
GSSNR	Global Sigma Signal To Noise Ratio
HS	Histogram Similarity
IF	Image Fidelity
LMSE	Laplacian Mean Square Error
MSE	Mean Square Error
NMSE	Normalised Mean Square Error
PSNR	Peak Signal To Noise Ratio
SC	Structural Content
SNR	Signal To Noise Ratio
Rot	Rotational Attack
Avg	Averaging Attack
Circ Avg	Circular Averaging Attack
Freq Filt	Frequency Filtering Attack
Gam Corr	Gamma Correction Attack
Gaussian	Gaussian Attack
Hist eq	Histogram Equalization Attack
Laplacian	Laplacian Attack
Nonlin Filt	Non Linear Filtering Attack
Resize	Resize Attack

Average Absolute Difference (AAD) based feature: This feature gives the degree of similarity of individual pixel under consideration. AAD is defined as

$$AAD = \frac{1}{MN}\sum_{m,n}|I_{m,n} - I'(m,n)| \tag{3.4}$$

where $I_{m,n}$ and $I'_{m,n}$ are the pixels at (m,n) location of a frame in original and target video respectively.

Global Sigma Signal to Noise Ratio (GSSNR) based feature: This feature is also used in comparing images like [63]. GSSNR can be defined as

$$GSSNR = \sum_{m,n}\frac{\sigma_b^2}{\sum_b(\sigma_b - \sigma'_b)^2} \tag{3.5}$$

where $\sigma_b = \sqrt{\frac{1}{p}\sum_{blockb} I_{m,n}^2 - (\frac{1}{p}\sum_{blockb} I_{m,n})^2}$ where $I_{m,n}$ and $I'_{m,n}$ are the pixels at (m,n) location of a frame in original and target video respectively.

Laplacian Mean Square Error (LMSE) based feature: This feature is chosen as this feature can describe the spatial error distribution across an image and thus the overall image quality can be computed by laplacian mean square error (LMSE) [124]. It can be defined as

$$LMSE = \frac{\sum_{m,n}(\nabla^2 I_{m,n} - \nabla^2 I'_{m,n})^2}{\sum_{m,n}(\nabla^2 I_{m,n})^2} \tag{3.6}$$

Mean Square Error (MSE) based feature This feature is chosen as it computes the average of the square of the errors to minimize the error due to randomness incurred in the watermarked video. MSE can be defined as

$$MSE = \frac{1}{MN}\sum_{m,n}(I_{m,n} - I'(m,n))^2 \tag{3.7}$$

where $I_{m,n}$ and $I'_{m,n}$ are the pixels at (m,n) location of a frame in original and target video respectively.

Peak Signal to Noise Ratio (PSNR) based feature: This feature is chosen as it gives a holistic comparison between two video frames. We have selected this feature as this is most commonly used feature to evaluate the video quality

$$PSNR = MN * max_{m,n}\frac{I_{m,n}^2}{\sum_{m,n}(I_{m,n} - I'(m,n))^2} \tag{3.8}$$

where $I_{m,n}$ and $I'_{m,n}$ are the pixels at (m,n) location of a frame in original and target video respectively.

Histogram Similarity (HS) based feature: This feature is used to get the degree of similarity of cumulative property of luminance and chrominance feature as a whole. HS can be defined as

$$HS = \sum_{c=0}^{255} |f_i(c) - f_i'(c)| \tag{3.9}$$

where $f_i(c)$ and $f_i'(c)$ are the relative frequency of level c for watermarked and attacked video frame respectively.

Image Fidelity (IF) based feature: This feature is chosen as it gives a holistic comparison between the two frames. It can be defined as

$$IF = 1 - \frac{\sum_{m,n}(I_{m,n} - I'_{m,n})^2}{\sum_{m,n} I_{m,n}^2} \tag{3.10}$$

where $I_{m,n}$ and $I'_{m,n}$ are the pixels at (m,n) location of a frame in original and target video respectively.

Normalised Mean Square Error (NMSE) based feature: NMSE can be defined as

$$NMSE = \frac{\sum_{m,n}(I_{m,n} - I'_{m,n})^2}{\sum_{m,n} I_{m,n}^2} \tag{3.11}$$

where $I_{m,n}$ and $I'_{m,n}$ are the pixels at (m,n) location of a frame in original and target video respectively.

Structural Content (SC) based feature: This feature is chosen as it can compare two frames even if one of the frame has under gone some affine transformation. SC can be defined as

$$SC = \frac{\sum_{m,n} I_{m,n}^2}{\sum_{m,n} I_{m,n}^2} \tag{3.12}$$

where $I_{m,n}$ and $I'_{m,n}$ are the pixels at (m,n) location of a frame in original and target video respectively.

Signal to Noise Ratio (SNR) based feature: This is one of the most common feature used to judge the video quality. SNR can be defined as

$$IF = \frac{\sum_{m,n} I_{m,n}^2}{\sum_{m,n}(I_{m,n} - I'm, n)^2} \tag{3.13}$$

where $I_{m,n}$ and $I'_{m,n}$ are the pixels at (m,n) location of a frame in original and target video respectively.

3.4.2.2 Defining the Mapping Between Observed Feature Value to Normalized Scale

These above mentioned feature values are not in a non uniform scale. So it is not possible to unify them to reach to a single decision. So we have used a mapping between the feature value to adjective factor. For each of the features we go through the following steps:

- Take 20 images and degrade them synthetically to five different levels
- 20 users (15 men and 5 women) are asked to judge degraded image based on their perception and to score them in a scale of 0 to 5 where 0 indicates the worst and 5 indicates the best quality
- Feature values for those 20 set of images are computed
- Define a mapping for the feature values to the scale of 5 which is purely based on human vision psychology from those images
- Thus we obtain a normalized scale where each feature can be mapped to a scale of 5 based on HVS
- For example, PSNR value is mapped to the normalized scale as below:

  $norm\_f_{PSNR} = 5$ if PSNR ≥ 36

  $= 4$ if PSNR ≥ 32

  $= 3$ if PSNR ≥ 28

  $= 2$ if PSNR ≥ 24

  $= 1$ if PSNR < 24

3.4.2.3 Assignment of Adjective Factor to Video Quality

- Compare the original video and the watermarked video frame by frame by using the above mentioned features
- Compute the average values of the i^{th} feature value avg_f_i as $avg\_f_i = \frac{1}{n}\sum_{k=1}^{k=n} f_{i,k}$ where n is the number of frame in the video.

- Apply the mapping between feature value avg_f_i to the normalized scale and thus we get a normalized score for the i^{th} feature as $norm\_f_i$
- Thus we get a column matrix where each row indicates the normalized feature value
- Now we define a mapping function that maps the m-dimensional vector into a one dimensional scalar. Such a function is called Additive Standard Multifactorial (ASM) function. Here we define the ASM as

$$value = (f_{AAD} + f_{GSSNR} + f_{LMSE} + f_{PSNR} + f_{MSE}) * 3 + f_{HS} + f_{IF} + f_{NMSE} + f_{SC} + f_{SNR} \tag{3.14}$$

- 20 users(15 men and 5 women) are requested to judge watermarked and video sequence based on their perception. This judgement is purely based on human vision psychology (HVS). All these opinions are summed up in Mean Opinion Score (MOS).
- An adjective factor is assigned to "VALUE" that matches the result obtained from HVS. The method used is like

  $video\_quality$ = "Excellant" if value ≥ 90

  = "Good" if value ≥ 80

  = "Average" if value ≥ 75

  = "Bad" if value ≥ 70

  = "Poor" if value < 70

3.4.3 Evaluation of the Robustness of the Algorithm

In this section we shall discuss about the details of the method to evaluate the robustness of the video watermarking method. Robustness is an important feature to judge a watermarking system. Robustness is measured by how the watermarking scheme withstand against attack. So it can be evaluated by how the hidden watermark information is retrieved after attacks. But some times the attack degrades the video below the acceptance level. So the robustness can be evaluated by considering two parameters:

- How the video sequence distorted by attack
- How well the embedded information is retrieved by watermark detector after attack

Table 3.9: Evaluation of Video Quality after Attack

Feature Name	RoA	RoA	AA	AA	CAA	CAA
	Feature	Norm	Feature	Norm	Feature	Norm
AAD	3.337	4.000	0.000	5.000	3.640	4.000
GSSNR	5.887	4.000	581629.059	5.000	5.747	4.000
HS	1453.551	5.000	0.000	5.000	3103.596	4.000
IF	0.973	3.000	1.000	5.000	0.996	5.000
LMSE	0.509	1.000	0.064	5.000	0.691	1.000
MSE	477.011	1.000	0.000	5.000	68.730	2.000
NMSE	0.027	2.000	0.000	5.000	0.004	5.000
PSNR	21.638	1.000	92.116	5.000	29.820	3.000
SC	0.975	2.000	1.000	5.000	0.981	3.000
SNR	16.039	1.000	90.000	5.000	24.822	2.000
value		48		100		66
Adjective factor		Poor		Excellent		Bad

3.4.3.1 Evaluation of Distortion in Video Quality After Attack

To evaluate the video quality after attack we apply the same methodology that is described in previous section. Here we perform the same set of operations to conclude on the video quality after attack by comparing the attacked stream with the watermarked stream. In Table 3.9 a set of factor values, their mapping to normalized scale and the decisive adjective factor is shown in tabular format for three attacks.

3.4.3.2 Evaluation of the Retrieved Watermark After Attack

Usually binary image, or any stream of text is used as watermark. So in this section first we shall discuss about the different parameters by which we can compare the retrieved text/image with the original watermark.

Evaluation of the retrieved binary image after attack: We have used the following features to compare two binary images

- Centroid deviation of 1s and 0s
- Bit error
- Crossing count feature

The method of evaluating the retrieved image quality after attack is described below:

Centroid deviation of 1s and 0s: Most of the cases binary images are used as watermark image. Here is the algorithm for this measure:

- Compute the Centroid of black pixels and white pixels for the image that has been used as watermark
- Compute the same for retrieved image
- Now find the deviation for black pixels and white pixels.
- Compute Euclidian distance of Centroid of black pixels and white pixels of retrieved and original binary image. Let the average deviation be denoted as (d)
- If the resolution of the embedded binary image is of height (h) and width (w), the deviation parameter (d_e) is computed as

$$d_e = \frac{d * 100}{\sqrt{h^2 + w^2}} \tag{3.15}$$

Bit error feature: Bit error (b_e) is the number of bits differing between retrieved and original binary image represented in percentage. So if there are m number of bits are differing in their values from the retrieved image and the watermark then the bit error is defined as

$$b_e = \frac{m * 100}{h * w} \tag{3.16}$$

where height and width of the image are specified by h and w respectively.

Crossing count feature: If image of some text is inserted as the watermark, number of 0 to 1 transition or 1 to 0 transitions is an interesting feature.

- Compute the 0 to 1 and 1 to 0 transitions for each row of original image
- Compute the 0 to 1 and 1 to 0 transitions for each column of original image
- If be the difference in crossing count of 0 to 1 of original and retrieved binary image, Crossing count error (c_e) is defined as

$$c_e = \frac{c * 100}{h * w} \tag{3.17}$$

- Error in retrieved image is defined as

$$e = \frac{c_e + d_e + b_e}{3} \tag{3.18}$$

As each of the factors can vary in a scale from 0 to 100, the value of e can also vary in the range of 0 to 100. Here, we have applied the similar MOS based approach as defined in earlier section. The mapping between the empirical value to adjective factor is defined below:

image_quality = "Excellant" if e $<$ 5

= "Good" if 10 $>$ e $\geq$ 5

= "Average" if 30 $>$ e $\geq$ 10

= "Bad" if 30 $>$ e $\geq$ 50

= "Poor" if e *ge* 50

Evaluation of the retrieved text after attack: We have used the following features to compare two text streams

- Hamming distance between texts
- Levenshtein distance

Now let us discuss these features one by one.

Hamming distance: In information theory, the Hamming distance between two strings of equal length is the number of positions for which the corresponding symbols are different. It measures the number of edit required to change one into the other, or the number of errors that transformed one string into the other. We represent the hamming distance in percentage and let the distance be h

Levenshtein distance: The Levenshtein distance or edit distance between two strings is given by the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character. It is useful in applications that need to determine how similar two strings are, such as spell checkers. It can be considered as a generalization of the Hamming distance, which is used for strings of the same length and only considers substitution edits. There are also further generalizations of the Levenshtein distance that consider, for example, exchanging two characters as an operation, like in the Damerau-Levenshtein distance algorithm. Compute levenshtein distance between the retrieved text and the watermarked text in percentage based score. We compute the mean error (t_e) as

$$t_e = \frac{l + d}{2} \tag{3.19}$$

where l is the Levensthein distance. As each of the factors can vary in a scale from 0 to 100, the value of t_e can also vary in the range of 0 to 100. Here, we have applied the similar MOS based

approach as defined in section III. The mapping between the empirical value to adjective factor is defined below:

text_quality = "Excellent" if $t_e < 5$

= "Good" if $10 > t_e \geq 5$

= "Average" if $30 > t_e \geq 10$

= "Bad" if $30 > t_e \geq 50$

= "Poor" if $t_e \geq 50$

3.4.3.3 Aggregation of the Above Three Methods for Single Point Decision Making

Architecture of the decision making process is shown in Figure 3.13 and the overall decision making metrics are given in the tabular format in Table 3.10. We conclude that the watermarking scheme is good if t_e and e is Excellent or good and VALUE is Good or excellent. If VALUE is Average or bad or poor, we conclude that the attack is not good because it losses the image quality significantly during attack.

In descriptive terms, the aggregation can be performed using the concept that if there is no significant degradation in video quality and the retrieved watermarked information does not contain significant errors, then the watermarking scheme has a high measure of goodness.

When we get all these features we come to decision about the quality of the watermarking based on Mean-Opinion-Score (MOS). We have used 20 users (15 men and 5 women) to judge attacked and original watermarked video sequence based on their perception. This judgment is purely based on human vision psychology (HVS).

3.5 Results and Discussions

We have tested the system against a video watermarking technique described earlier in Section 3.3. against 20 different standard teststreams downloaded from ITU-T website. Out of those 20 test streams 2 were used to obtain the MOS and rest 18 were used in testing. All the teststreams were tested against the suit of attacks already described. In this section we present the report obtained by applying some attacks of Stirmark on the described watermarking scheme.

Initially the watermarked video stream is compared against original watermarked video stream to evaluate the video quality. Then the video quality of the attacked video is compared against the watermarked video. Table 3.11 shows the scores for video quality obtained by applying

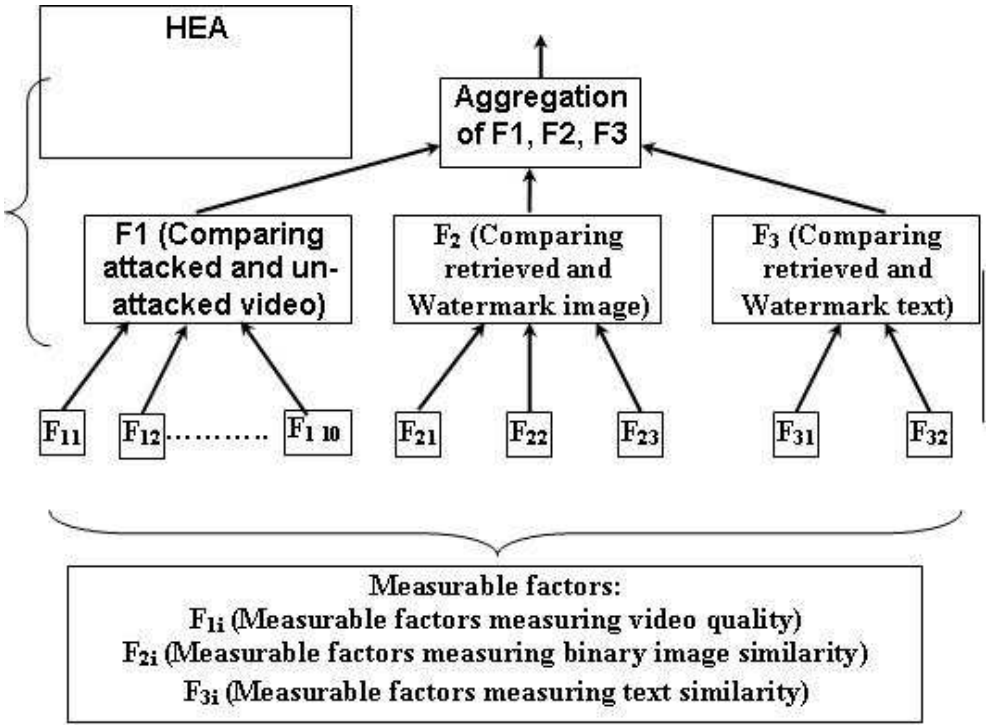

Figure 3.13: Decision Making System Architecture

Table 3.10: Overall Decision Making Process

Video Quality after	Error in retrieved attack	Error in retrieved image	Robustness Text
Any	Excellent	Excellent	Excellent
Any	Excellent	Good	Good
Any	Good	Any	Good
Any	Any	Good	Good
Excellent or Good	Medium	Medium	Medium
Excellent or Good	Bad or Poor	Medium	Bad
Excellent or Good	Medium	Bad or Poor	Bad
Excellent or Good	Bad or Poor	Bad or Poor	Poor
Medium, Bad or Poor	Medium, Bad or Poor	Medium, Bad or Poor	Attack degrades video quality beyond accepting criteria - hence attack is not suitable

different attacks on the watermarked video sequence and the decision made out of it. It shows that the video quality remains excellent after attack for averaging attack and resizing attack. In Figure 3.14 screen shots of attacked videos are shown along with the same frame in original video. From this figure it can be observed that the attacked video degrades most in case of FFA, NLFA, GCA, HEA, and LEA. In case of RoA, video quality remains almost same in the central part of the video but in the corner a huge amount of degradation can be found. This degradation impacts the visual experience by the user. It is also supported by the error values obtained by using the parameters described in the proposed methodology. Those error values are shown in table 3.11. So it can be concluded that the parameters selected by us for comparing the video quality gives a quantitative measure of Human Vision Psychology as the decision made based on those parameters gives the similar result obtained from human perception. Table 3.12 and Table 3.13 give the Retrieved Image and Text quality metrics for each of the attacks under consideration, respectively. The retrieved logos from the attacked videos are also shown in Figure 3.14. It shows that the retrieved logo can not be visually recognized in case of FFA, but rest of the attacks can not destroy the logo information. Similar results obtained quantitatively is shown in Table 3.12. So we can conclude that the features selected by us for comparing the retrieved binary images gives the similar results obtained from human perception.

Table 3.14 combines the results described above to conclude on the overall performance of the Watermarking Algorithm to evaluate. This decision is taken based on the overall decision making system described in Table 3.10. The extracted watermark (text and logo) is very similar to the inserted watermark against AA, GCA, HEA, LEA, RoA, and RsA though the attacked video degrades bellow the acceptance limit in case of GCA, HEA, LEA, and RoA. So we conclude that the watermarking method is very robust against those attacks. But we cannot conclude in case of FFA and NLFA as the attacked video quality is drastically degraded.

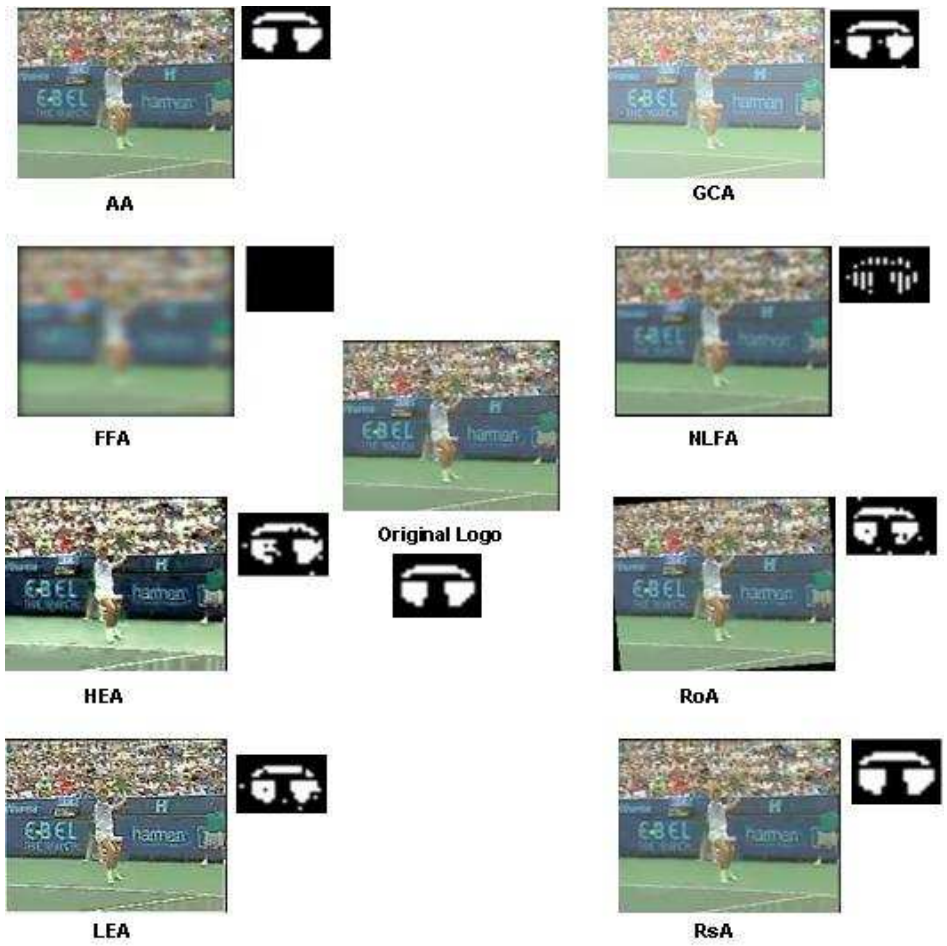

Figure 3.14: Screen Shots of Frames of Original and Attacked Video and Retrieved Watermark

Table 3.11: Video Quality after Attack

Attack	V_e	C_{qual}
AA	100	Excellant
FFA	25	Poor
GCA	27	Poor
HEA	27	Poor
LEA	28	Poor
NLFA	25	Poor
RsA	100	Excellant
RoA	37	Poor

Table 3.12: Retrieved Watermark Image Quality after Attack

Attack	b_e	c_e	d_e	I_e	Image Quality (C_{Img})
AA	0.000	0.000	0.000	0.000	Excellent
FFA	5.469	10.938	55.172	23.860	Poor
GCA	0.781	1.563	3.448	1.931	Good
HEA	1.563	1.563	3.448	2.191	Good
LEA	1.823	2.083	0.000	1.302	Good
NLFA	5.729	10.417	13.793	9.980	Medium
RsA	0.000	0.000	0.000	0.000	Excellent
RoA	0.781	0.521	0.000	0.434	Excellent

Table 3.13: Retrieved Text Quality after Attack

Attack	l	H	Te	Text Quality (C_{txt})
AA	0	0	0.000	Excellent
FFA	6	1	3.5	Bad
GCA	0	1	.5	Good
HEA	6	7	6.5	Poor
LEA	4	5	4.5	Bad
NLFA	6	1	3.5	Bad
RsA	0	0	0.000	Excellent
RoA	0	0	0.000	Excellent

Table 3.14: Results and Conclusions

Attack	Video Quality (C_{qual})	Image Quality (C_{Img})	Text Quality (C_{txt})	Robustness against Attack
AA	Excellent	Excellent	Excellent	Excellent
FFA	Poor	Poor	Bad	Attack Degrades Video Quality
GCA	Poor	Good	Good	Good
HEA	Poor	Good	Poor	Good
LEA	Poor	Good	Bad	Good
NLFA	Poor	Medium	Bad	Attack Degrades Video Quality
RsA	Excellent	Excellent	Excellent	Excellent
RoA	Poor	Excellent	Excellent	Excellent

Chapter 4

Mash up of Textual context of broadcast video and Web Information

4.1 Introduction

Recent market trends on consumer electronics show that consumers are demanding for Internet connected TV largely as the sale for such product raises high in the second quarter of 2009 compared to the first quarter of the same year [125] - [126]. One such product that connects a television set with the internet can be found from [127]-[128]. Gartner report also suggests that the new Connected TV widget-based services are quite impressive [129].

The survey on the wish list of the customers of connected TV shows that there is a demand of a service where the user can get some additional information, from Internet or different RSS feeds, related to the news show that the customer is watching in TV. Features of existing Connected TV solutions are well described in [130]. A comprehensive analysis on the pos and cons of the products on Connected TV can be found from [131], [132]. But none of the features in those literatures can meet this requirement. Only one nearly similar feature is given by Microsoft as shown by them in International Consumer Electronics Show (CES) 2008 where the viewers can access the contents on election coverage of CNN.com while watching CNN's television broadcast, and possibly even participate in interactive straw votes for candidates [133]. But this is strictly applicable when the customer is watching some specific news channel only. Moreover this solution is based on STBs or IPTVs. But the report from Telecom Regulatory Authority of India (TRAI) reveals that even in 2008 only 6.55% of the total the number Houses owning a TV are using Direct to Home (DTH) service. So the above survey reflects that there is no existing solution that can work on TVs using the RF feed as input.

So there is a definite need for a system that can recognize the breaking news content automatically and can provide the related information from different RSS feed or Internet that is

independent of the source video i.e the video may come from either DTH or RF cable.

So in this chapter an end to end system is proposed that can bridge the gap between technology and market demand. In the proposed method initially the text regions of the video are localized. The content for each of the regions containing the text information is then recognized. Next some heuristics based key word spotting algorithm is applied. These heuristics are purely based on the observation on the breaking news telecasted in Indian news channels. Finally we have also proposed a method of data representation so that users can use that information effectively and efficiently.

In this chapter first an overview of the proposed system is given and then the steps involved in the work are described in different sections.

4.2 Recognition of Textual Context: Some Use Cases

Some use cases are described in a work related to the thesis [134].

4.2.1 Automatic Classification of Videos for Storage/Retrieval

There are different kinds of programs that are broadcast on TV - News, Sports, Entertainment/Movies etc. It would be of immense help if these videos can be classified automatically based on the content. Possible use cases may include offline recording of TV programs for later analysis and Digital Video Recording (DVR). While each of these videos may have certain underlying properties (Sports Videos are more fast-changing, News Videos contain more static scenes etc.), it is really very difficult to classify the videos reliably in that way. A far more interesting and useful approach lie in trying to detect the texts that are coming embedded inside such videos. Once the texts are detected, the very semantics of the text can give enough clues on the type of the Video. For example, detection of a "Breaking News" or Stock Tickers would characterize a News Channel Figure 4.1. Detection of "Score" type text can signify a Sports channel (Figure 4.2). Presence of sub-titles can signify a movie channel. Once the videos are classified, it becomes easy to store them with the classification. This in turn results in a faster and easier retrieval mechanism.

4.2.2 Advertisement Removal for News Video/Movies

Advertisements can also be characterized by the text embedded in them (Figure 4.3). Once they are detected, they can easily be removed during storage.

Figure 4.1: TV Video Depicting Keyword Texts in News Video

Figure 4.2: TV Video Depicting Keyword Texts in Sports Video

Figure 4.3: Screen Shots Depicting Advertisement

Figure 4.4: TV Video Depicting Keyword Texts in News Video

4.2.3 Automated Video Indexing for Internet Search

Currently Video Indexing for Search is normally based on the filenames. For example, if the search query is "Rose", then it will try to find all files that contains rose in the filename of the image file. However, texts embedded in the videos can provide far more information about the context of the video. If text recognition engine is run across the complete video, it can yield a set of keywords that best describes the context of the Video (Figure 4.4). The complete Search Indexing can be based on these Keywords. Again, this may not be an independent technology, but a complementary one that works hand-in-hand with existing Video Indexing Technologies employed to create far more accurate multi-modal implementations.

4.2.4 Duplicate News Story Detection

News broadcasts are often accompanied by text tickers that describe the corresponding news event in the video. In applications where video recording is required for multiple TV channels, one can identify duplicate News Stories occurring in multiple channels/repeat broadcasts by just recognizing the accompanying text and comparing the recognized text for duplication (Figure 4.5, Figure 4.6). This could be a way by which one can avoid the duplication in recording and save storage space. One such solution is described in [135].

4.2.5 Personalized Stock Market Ticker

Normally Stock tickers are embedded into TV broadcast as a continuous stream consisting of all stocks. However, a user may have interest in only a few particular stocks. OCR can be employed

Figure 4.5: TV Screen Shots Depicting Same news in Two Channels (in CNN)

in real-time on the stock ticker section of the TV broadcast video and only the stocks of interest can be picked. The picked stock names and their prices can be shown as a separate text overlay ticker on top of the current TV Video. The Set top Box giving feed to the Television needs to have overlay feature in order to enable this functionality.

4.2.6 Personalized Mash-up of the Internet News with TV News

As an extension of the same technology described in [136], TV news broadcast normally contains some "Breaking News", text tickers. These texts can be obtained by applying OCR on those text regions. Once the OCR recognizes the news texts, it can be converted to a set of keywords with help of a pre-defined dictionary. These keywords can then be used to fetch related news from Internet by subscribing to different Internet News feed channels. This results in a stream of news information that is contextually related to the news video currently broadcast on TV. The whole process is depicted in Figure 4.7. This news information stream can either be overlaid on top of current TV video or can be stored inside the Set top Box for later access by the user. This work is described in [136].

4.3 System Overview

Figure 4.8 depicts an overview of the system of web and contextual TV information mash up. Broadcast TV content is fed into the video capture module which then enters the post processing sub-system via the context analysis interface. The video context is analyzed in three levels: text, channel logo and known objects. In this section we shall discuss about the text information

Figure 4.6: TV Screen Shots Depicting Same news in Two Channels (in Times Now)

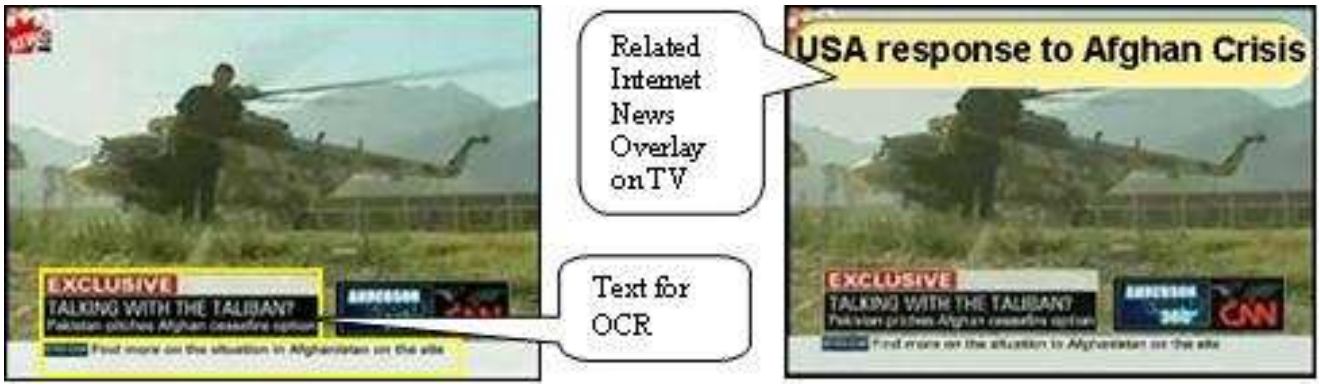

Figure 4.7: Intelligent Mash-up of TV News with Internet News

extraction only. This analysis generates an array of context information for the video frame. This array is fed to the application along with the TV video. The application shall use this context data to get relevant mash-ups from Internet and then display the same along with TV content. Steps required for keyword spotting from the news video is shown in Figure 4.9.

As the first step, we localize the text regions present in the news video frame. The proposed system consists of the following items:

- Input to the system comes in either digital format (in case of DTH services) or in form of analog TV signal with video and audio (in case of RF cable)
- In case of analog TV feed, it is converted to a raw video in YUV 4:2:2 format
- Text region are identified from the video
- The candidate text regions are processed
- Breaking news are identified from the recognized texts
- Fetch information related to the breaking news from internet or RSS feeds
- Display by mashing up the internet information with the TV video

4.4 Localization of Textual Region

In the earlier section it was mentioned that the input for the proposed system can either be a digital feed (in case of DTH) or an digitized version of the analog video feed (in case of RF feed cable). In section 1.2.3 it was mentioned that two different approaches for Text Information Extraction (TIE) is possible. So in case of the RF cable based input a low cost pixel based TIE is proposed as the video signal is in raw YUV or RGB format after getting digitized. On the other hand in case of DTH, video input comes as H.264 compressed video format. So for those cases a compressed domain approach for TIE is used. But one limitation of that approach is that it produces quite a good number of false positives. So a hybrid approach is also proposed where the candidate text regions are initially localized using the compressed domain features without any significant coding overhead and then the pixel based approach is used to confirm the text regions.

Our proposed methodology is based on the following assumptions based on the observation from different sports sequences.

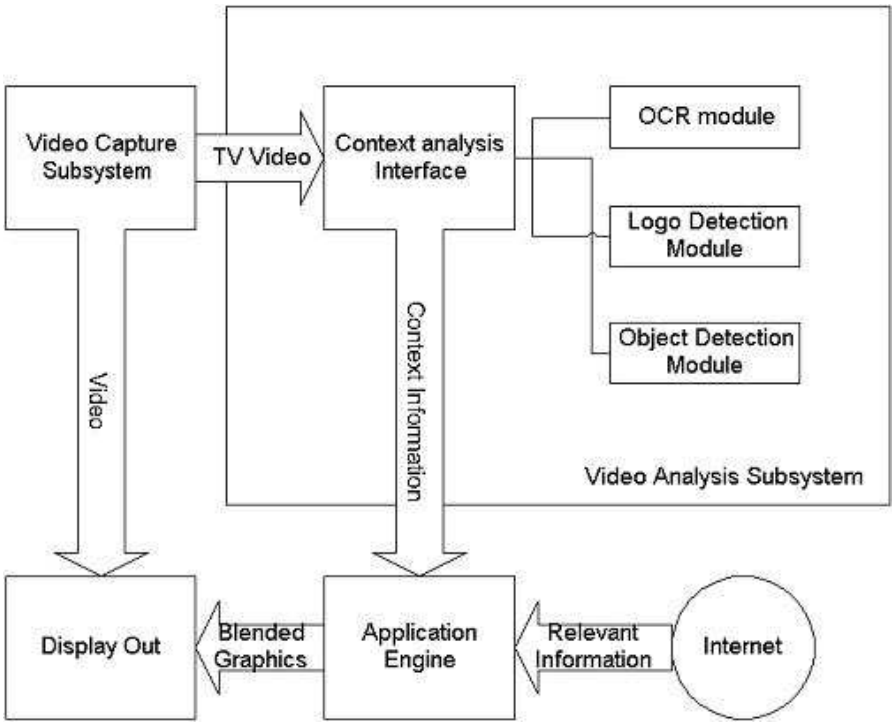

Figure 4.8: System Overview

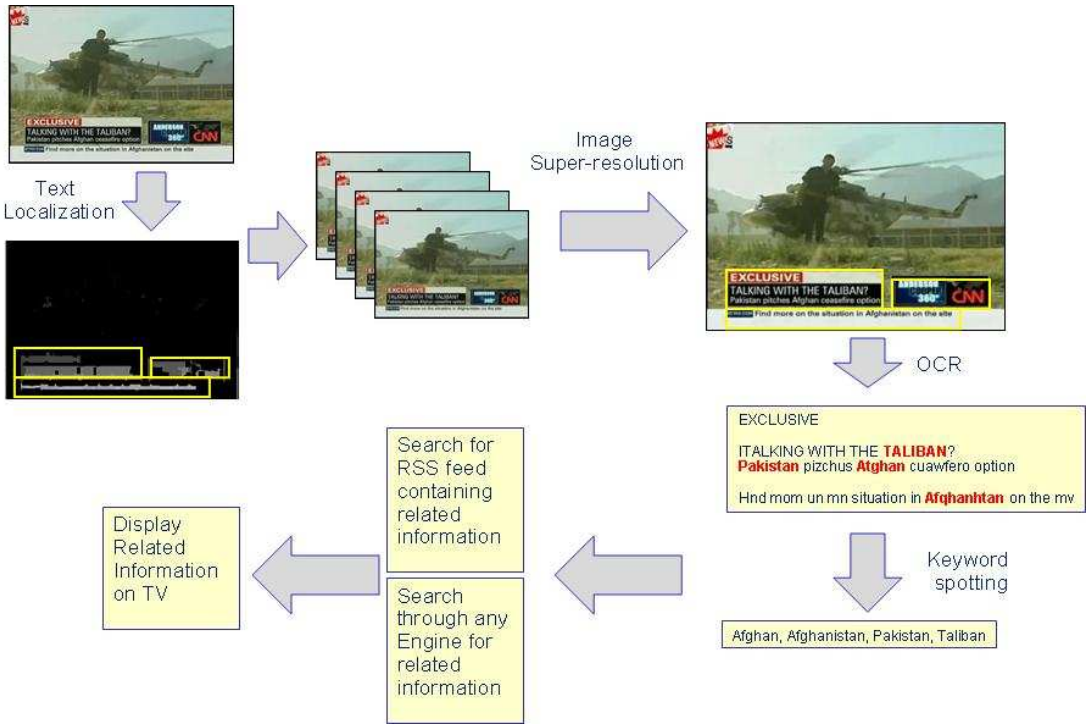

Figure 4.9: Steps Involved in the System

- Text regions should have a high contrast
- Texts should be aligned horizontally
- The components representing texts region has a strong vertical edge
- Trademarks persists in the video for at least 0.2 seconds
- Some of the trademarks are not perceptible. We shall consider only the perceptible texts in the video

4.4.1 Using Pixel Based Approach

In this module the the suspected text regions are localized. This module is based on the assumptions already mentioned. This similar approach is used in real time realization of another application as described in [137].

4.4.1.1 Filter Out the Low Contrast Components

Initially all the low contrast components from a video are filtered out as we assumed that the text regions are of high contrast. This assumption is very obvious as the sponsors are usually spending dollars for getting visibility and human eye is much sensitive in high contrast regions as compared to the low contrast one.

In this realization, we have assumed that the input raw video is in YUV format. As human eye is more sensitive to luminosity information than the color, we have used only Y component of the video in TIE. Here we compute the difference of intensity (I_{diff}) of the neighbouring pixels and then mark all pixels for which $I_{diff} > \tau_{contrast}$ to get the binarized output video (V_{cont})

$$I_{diff} = abs(I_i - I_{i-1}) \quad \forall i = 2, ..., n \tag{4.1}$$

Where I_i is the intensity of the i^{th} pixel, n is the width of the video frame and $\tau_{contrast}$ is a statistically obtained value. Two typical video frame and the high contrast regions in those frames are shown in Figure Figure 4.10.

4.4.1.2 Morphological Closing

Because of the video quality, in V_{cont} the text components are not getting disjoint. Moreover the non textual regions are also coming as noises. So a morphological closing operation is applied

on V_{cont} to get a morphed video frame V_{morph}

$$V_{morph} = Dilate(erode(V_{cont})) \tag{4.2}$$

V_{morph} for the same set of frames is shown in Figure 4.11. In this application we have used a rectangular structural element with dimension of 3x5.

4.4.1.3 Confirmation of the Text Regions Using Shape Feature

This is the method to remove the non textual regions from the video frame based on the assumptions. The pseudo code for removing the non textual part is as below:

- Run connected component analysis for all $P_c \in V_{morph}$ to split the candidate pixels into n number of components (c_i) where P_c is a pixel in V_{morph}
- Find the area for each c_i
- Remove the components with area smaller or greater than two experimentally obtained threshold values.
- Remove all components for which compactness $compactness > 1.0 \quad Or \quad compactness < 0.2$ Where $compactness = \frac{PixelCount}{Area}$
- Find the bounding box for each components for all remaining components.
- Compute the mode for x and y coordinate of top left and bottom right coordinate (tl_x, tl_y, br_x, br_y). Let the top left coordinate for the i^{th} component be represented as pos_i
- Find the threshold (τ) for removing non textual components as

  $$\tau = mode(median(pos_i) - pos_i) \tag{4.3}$$

  here we have used Euclidian distance the distance between two coordinates.
- Mark all c_i for which $(median(pos_i) - pos_i) < \tau$ to get a video frame containing only candidate text regions (V_{cand})

4.4.2 Compressed Domain Processing

Our proposed method is based on the following two features which can directly be obtained during decoding of compressed domain H.264 stream. This algorithm was used to detect the billboards of soccer and cricket ground as described in related work to this thesis [138].

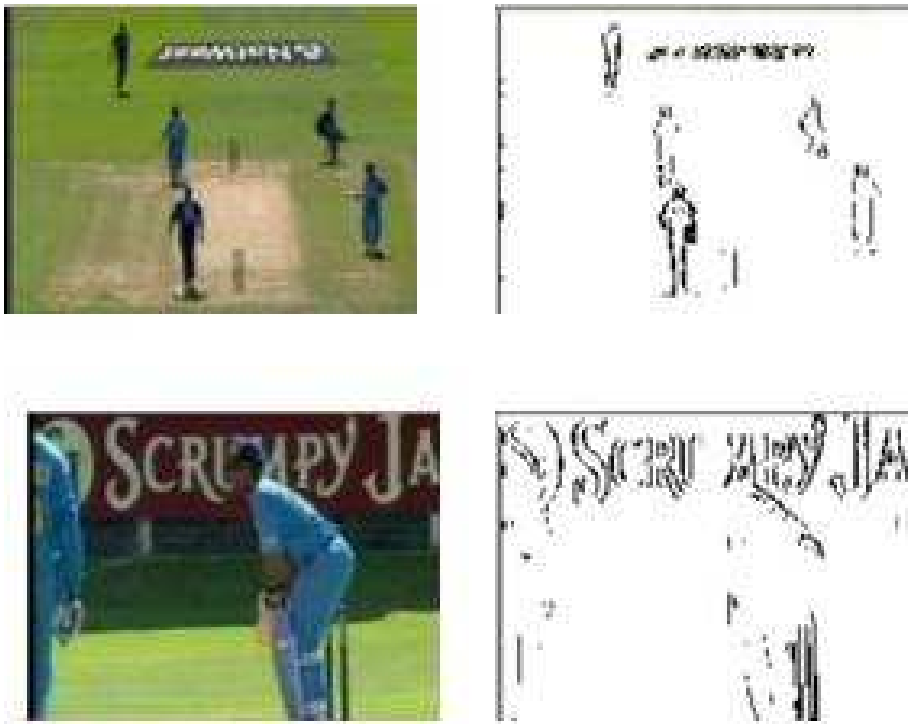

Figure 4.10: Two Typical Video Frame

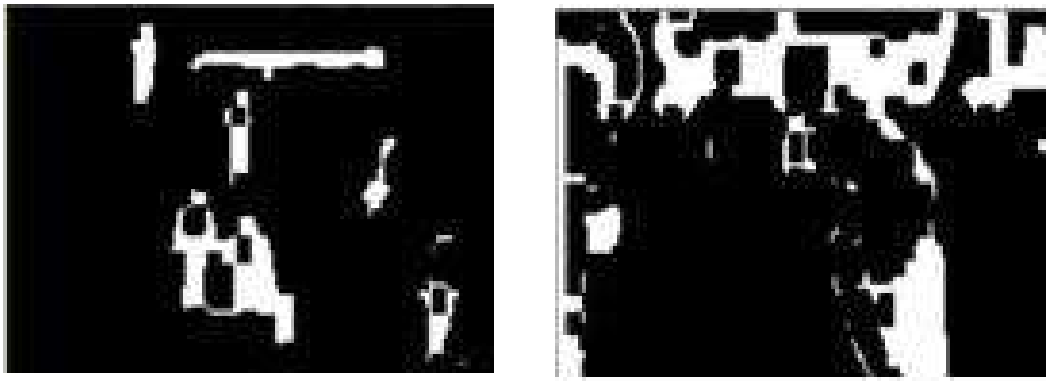

Figure 4.11: Two Typical Video Frame after Morphological Operations

4.4.2.1 DC Component of Transformed Luma Coefficient Based Features

In H.264 4x4 Integer transformation is used which is different from the 8x8 DCT transformation of MPEG series video codec. DC components of the integer transformed luma coefficients is a representative of the original video at a lower resolution. In H.264, unlike previous video codecs, 4x4 block size is used. So it gives the iconic representation of the video more precisely. The pseudo code of the algorithm in the proposed method is given below:

- Get the Luma DC value (dc_l) for each 4x4 sub block from decoder
- Compute the first order difference ($\partial_x(dc_l)$ and $\partial_y(dc_l)$) of dc_l with neighboring sub blocks in x and y direction.
- From observation it is found that the difference of DC value of two neighboring sub-block is very high for a high contrast region.
- As the problem is a two class problem, where each sub-block is either a text or is a non text, K-Means algorithm, with K = 2 is used.
- Find the centroid (τ_{dc}) of the high valued cluster.
- If $\partial_x(dc_l)$ or $\partial_y(dc_l)$ is greater than the experimentally obtained threshold value (τ_{dc}) mark that Macro block (MB) as a candidate text in this frame and store this MB number in an array (a_l)

4.4.2.2 De-blocking Filter Based Feature

Based on our observations on different sports videos, it is found that the trademarks and the logos are displayed with high contrast difference from the background. As a consequence the texts results in a strong edge at the boundaries of text and background. But in this approach an additional time complexity is required for detecting edges. One of the new features of H.264 which was not there in any previous video CODEC is deblocking (DB) filter. We have used the edge information extracted from the decoder during the process of decoding without computing the edges explicitly. A conditional filtering is applied to all 4x4 luma block edges of a picture, except for edges at the boundary of the picture and any edges for which the DB filter is disabled by disable_deblocking_filter_idc, as specified in the slice header. This filtering process is performed on a MB after the picture construction process for the entire decoded picture. For each macroblock and

each component, vertical edges are filtered first. The process starts with the edge on the left-hand side of the macroblock proceeding through the edges towards the right-hand side of the macroblock in their geometrical order. Then horizontal edges are filtered, starting with the edge on the top of the macroblock proceeding through the edges towards the bottom of the macroblock in their geometrical order. The pseudo code for selecting candidate frames using this feature is given below:

- Get the strength of DB filter for each MB
- If it is a strong vertical edge [11], mark that MB as a candidate.

4.4.2.3 Identifying the Text Regions

The pseudo code for removing the non textual part is as below:

- For each candidate MB, identify the X and Y coordinate top left position for each MB (c_x and c_y)
- Find the frequency (f_r) of candidate MB in each row.
- Remove all MBs from a_l If $f_r < 2$ where a_l is the array described in earlier in section 1.4.2.1.
- Check for continuity of MBs in each row: For this check the column number (c_x) for candidate MBs in a row.
- If $(c_x(i+1) - c_x(i)) > 2$ unmark the MB from a_l where $c_x(i)$ is the column number for i^{th} candidate MB in a particular row
- To ensure that time domain filtering we store one frame into buffer and display the i^{th} frame while decoding the $(i-1)^{th}$ frame.
- Unmark all candidate MBs in i^{th} frame if there is no candidate MB in adjacent $(i-1)^{th}$ frame and $(i+1)^{th}$ frame.
- Finally all marked candidates MBs are decided as Text content in the video.

4.4.3 Using Hybrid Approach

In this approach the advantage of both the pixel based approach and compressed domain approach is taken into account. In this approach, the candidate text regions are localized using compressed domain features which is not computationally expensive. But the problem with compressed domain approach is that it gives some false positives. So in turn the pixel domain features for those candidate text regions are used to remove those false positives. The main drawback of the pixel domain approach is that it is computationally expensive. But as only the candidate text regions are used for performing pixel domain operation, the computational cost is less.

4.4.3.1 Confirmation of the Text Regions

Initially we run a connected component analysis for all pixels after morphological closing to split the candidate pixels into n number of connected components. Then we eliminate all the connected components which do not satisfy shape features like size and compactness . Then we compute the mode for x and y coordinate of top left and bottom right coordinates of the remaining components. We compute the threshold as the mode of the difference between the median and the position of all the pixels. The components, for which the difference of its position and the median of all the positions is less than the threshold, are selected as the candidate texts. We have used Euclidian distance as a distance measure.

4.4.3.2 Confirmation of the Text Regions Using Temporal Information

At this stage, the text segments have been largely identified. But, some spurious segments are still there. We use heuristics to remove spurious segments. Human vision psychology suggests that eyes cannot detect any event within 1/10th of a second. Understanding of video content requires at least 1/3rd of a second, i.e. 10 frames in a video with frame-rate of 30 FPS. Thus, any information on video meant for human comprehension must persist for this minimum duration. It is also observed that the noise detected as text does not generally persist for significant duration of time. Thus, we eliminate any detected text regions that persists for less than 10 frames. At the end of this phase, we get a set of groups of frames (GoF) containing ticker text. The information together with the coordinates of the bounding boxes for the ticker text are recorded at the end of this stage of processing.

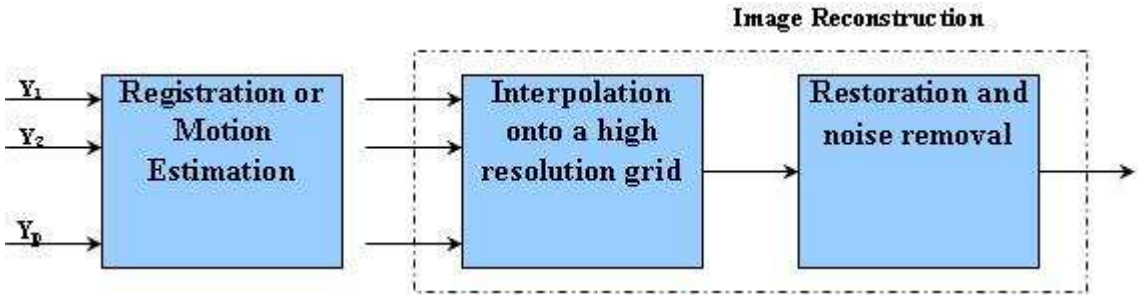

Figure 4.12: Stages of Image Super Resolution

4.5 Text Recognition

In this section two major steps are involved namely text region preprocessing and Optical Character Recognition (OCR).

4.5.1 Preprocessing

The GoF containing ticker text regions cannot be directly used with OCR software because the size of the text is still too small and lacks clarity. Moreover, the characters in the running text are often connected and need to be separated from each other for reliable OCR output. To accomplish this task we interpolate these images to a higher resolution by using Image Super Resolution (SR) techniques. The processing steps are given below:

4.5.2 Image Super Resolution (SR)

Figure 4.12 shows different stages of a multi-frame image SR system to produce an image with a higher resolution (X) from a set of images (Y_1, Y_2, Y_p) with lower resolution. We have used SR technique presented in [139] where information from a set of multiple low resolution images is used to create a higher resolution image. Hence it becomes extremely important to find images with same text. We perform pixel subtraction of both the images in a single pass. We now count the number of non-black pixels by using intensity scheme (R, G, B) $<$ (25, 25, 25). We then normalize this count by dividing it by total number of pixels and record this value. If this value exceeds statistically determined threshold β, we declare the images as non identical otherwise we place both the images in same set. As shown in Figure 4.12, multiple low resolution images are fed to an image registration module which employs frequency domain approach and estimates the planar motion which is described as function of three parameters: horizontal shift (δx), vertical

shift (δy) and the planar rotation angle (ϕ). In Image Reconstruction stage, the samples of the different low-resolution images are first expressed in the coordinate frame of the reference image. Then, based on these known samples, the image values are interpolated on a regular high-resolution grid. For this purpose bicubic interpolation is used because of its low computational complexity and good results.

4.5.3 Touching Character Segmentation

Once the binarized image is obtained very frequently it is observed that the image consists of a number of touching characters. These touching characters degrade the accuracy rate of the OCR. Hence the Touching Character segmentation is required to improve the performance of the OCR. Here is the pseudo code for the same

- Find the width of each character. It is assumed that each connected component with a significant width is a character. Let the character width for the i^{th} component be WC_i
- Find average character width $\mu_{WC} = \frac{\sum_{i=1}^{i=n} WC_i}{n}$ where is the number of character in th ROI
- Find the Standard Deviation of Character Width (σ_{WC}) as $\sigma_{WC} = STDEV(WC_i)$
- Define the threshold of Character Length (T_{WC}) as $T_{WC} = \mu_{WC} + 3 * \sigma_{WC}$
- If $WC_i > T_{WC}$ mark the i^{th} character as candidate touching character
- The number of touches in i^{th} candidate component is computed as $n_i = \lceil \frac{WC_i}{T_{WC}} \rceil + 1$
- Divide WC_i in n_i equally spaced segments

This processed image is given as the input for open source OCR module Tessaract. This OCR performs well in case of texts with larger font sizes.

4.6 Keyword Selection

In the proposed approach we have identified the keyword to search is based on heuristics which are in turn are based on some observations like

- Breaking news always comes in Capital Letter.
- Font size of those important news are larger than that of the ticker text

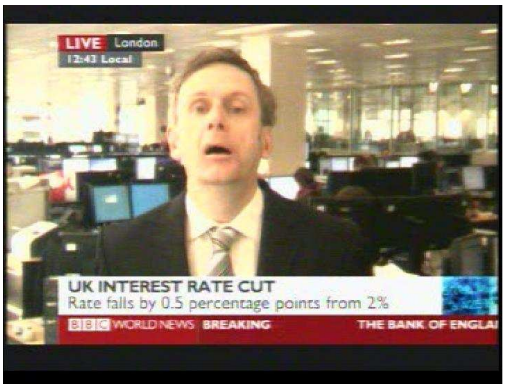

Figure 4.13: Screen Shots Showing that the Breaking News are in Capital Case from BBC

- They appear on either just above or just bellow the central region, where the anchor, studio or some news clippings are shown.

These assumptions can be justified by the screen shots of News shows telecasted in different news channels as shown in Figure 4.13- Figure 4.16.

From these observations we have used the following approach to identify the keywords as:

- Select the recognized words coming from the output of OCR with all capital letters
- Find the number of words in a text line
- If the number of words in a text line is above a heuristically obtained threshold value we consider them as candidate text region.
- If multiple such text lines are obtained find whether there is any such text line above the middle of the video frame
- If multiple such text lines are obtained below the middle of the video frame select the text line which is nearer to the middle of the video frame as the candidate text
- Remove the stop words (like a, an, the, for, of etc) from them Concatenate the remaining words to generate the search string for internet search engine
- Use this text line as the input to search the RSS feeds

selected keyword can be given to Internet using Web 2.0 APIs to fetch News Feeds and RSS feeds. The feed information fetched can be blended on top of TV video to create a mash up between TV and Web.

Figure 4.14: Screen Shots Showing that the Breaking News are in Capital Case from Times Now

Figure 4.15: Screen Shots Showing that the Breaking News are in Capital Case from CNN

Figure 4.16: Screen Shots Showing that the Breaking News are in Capital Case from NDTV 24x7

But in this chapter we shall mainly focus on two problems namely (i) Mash up of web information with contextual textual information and (ii) Automatic sending of SMS for Active page services of DTH.

4.7 Results and Discussions

4.7.1 Testing Environment

In this section, we shall describe the testing environment for all three types of applications we are discussing here.

This application has a number of modules like text localization, text recognition using OCR, Keyword spotting and information extraction (IE) from web. So any error introduced at any level gets accumulated for the next layer and thus the accuracy rate degrades. We have tested each of these modules individually in the PC based environment. They are manually annotated and kept as ground truth. The video of different news channel video is initially recorded and the performance of the algorithm is tested against those sequences.

Text localization module is tested against a set of test videos containing rich textual contents. We have identified five such categories of videos where textual content is very high like:

- News videos. Some such examples for different languages are shown in Figure 4.16, Figure 4.17, Figure 4.18.
- Recipe shows. Some such examples are shown in Figure 4.19, Figure 4.20, Figure 4.21.
- Sports videos with score and bill boards. One such example is shown in Figure 4.22.
- Active pages and Guide pages of DTH services. Two such examples are shown in Figure 4.23, Figure 4.24.
- Movies with subtitles. One such example is shown in Figure 4.25.

4.7.2 Method for Testing

We have not tested the performance against any of the benchmark video sequence of TREK as we have found that the video quality for the analog feed TV channels is much poorer than that of the TREK video. Moreover there is no guideline of video quality in terms of PSNR for India can be found from the literature. The testing method for this module is as follows:

Figure 4.17: Screen Shot News Video in Hindi

Figure 4.18: Screen Shot News Video in Bengali

Figure 4.19: Recipe Show in English

Figure 4.20: Recipe Show in Gujrati

Figure 4.21: Recipe Show in Bengali

Figure 4.22: Sports Video with Text

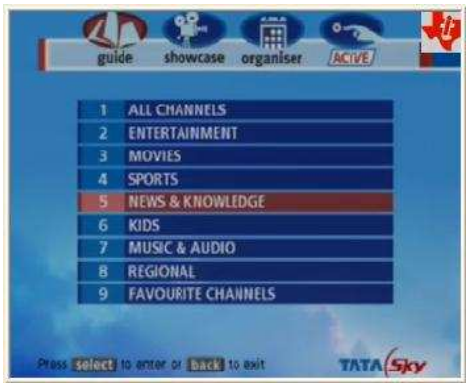

Figure 4.23: Guide Page

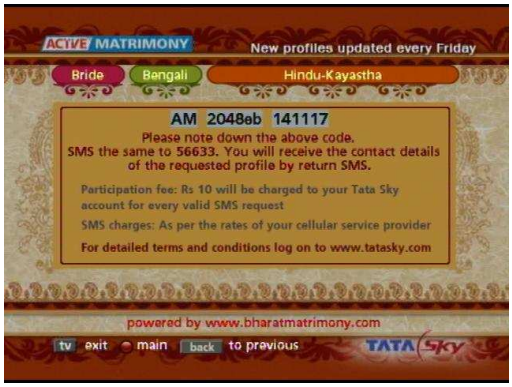

Figure 4.24: Active Page

Figure 4.25: Subtitle

- Record 20 news videos of 10 minute duration from different genres containing text information. So total 200 number of videos were used as the corpus for testing.
- Mark 20 such videos manually to mark the text regions.
- As the proposed approach is not using any machine learning methods and mainly based on some heuristic thresholds, we tune those thresholds and parameters for those 20 news videos.
- Test the localization accuracy based on the rest 180 news videos.
- Once the text regions are marked, they are given as a input for the OCR module. OCR recognizes the texts for those regions only.
- If there is any false positive in text localization, OCR returns a set of junk and meaningless character sequence for them
- We perform a special character count based threshold to remove those false positives.
- The character level accuracy is not analyzed as our prime interest is in getting the related web information accurately.
- The OCR output is given as an input to Google search engine and it was found that most of the character level wrong recognition is checked by the suggestion from Google.
- The pages suggested by Google are satisfying for the users and we didn't test the page recall rate explicitly.

4.7.3 Performance Evaluation and Discussion

The performance of each of the modules will be described in this section in terms of accuracy of recognition that can be measured using two parameters namely recall and precision and time to execute. Recall (r) and precision (p) can be described as below

$$r = \frac{c}{c+m}, p = \frac{c}{c+fp} \tag{4.4}$$

Where c is number of correct recognition of text regions, p is the number of misses, and fp is number of false positives.

Table 4.1: Comparative Performance Analysis of Different Text Localization Methods

Method	Recall	Precision	Time
Compress Domain	0.94	0.67	180
Pixel Domain	0.96	0.85	260
Hybrid	0.94	0.78	200

Proposed solution has a number of sub modules like text localization, Text recognition and required information obtained from the pages returned by Google search engine on the selected key words.

4.7.3.1 Text Localization Module

This module works with a recall rate of 0.94 but precision is 0.78. Performance of different approaches of text localization algorithm and their relative time requirement (in mili second per frame) is depicted in Table 4.1. One screen shot showing the original video frame and corresponding output is shown in Figure 4.26 and Figure 4.27 respectively.

The reasons behind misses are as follows:

- Whenever there is no high difference of intensity between the foreground and the background of the text.
- Proposed text localization module can not perform well whenever the font size is too small.
- Whenever there is a text that is not aligned horizontally the proposed method can not recognize it as we have made an assumption that the texts are aligned horizontally.

Figure 4.26: Screen Shots of Original Video Frame

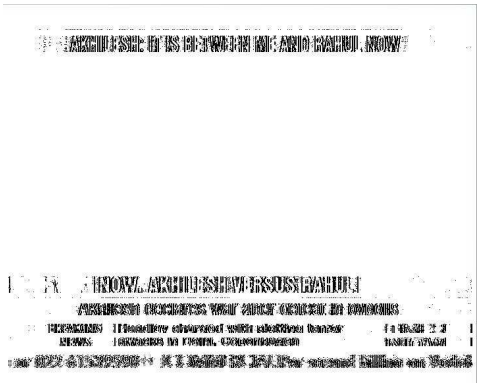

Figure 4.27: Screen Shots of Output Video Frame

The reason behind a low precision rate is that we have tuned the parameters and threshold values in a manner so that the probability of false negative (misses) is minimized. Moreover we have a check after getting the output from the OCR module as described earlier.

4.7.3.2 Accuracy in Text Detection

Once the text regions are located each candidate text rows undergo some processing prior to OCR and are given as input to Tessaract for OCR. It is found that in case of false positives a number of special characters are coming as output of OCR. So we discard the candidate texts satisfying the following condition:

$$specialcharacter/alphabetratio > 1 \tag{4.5}$$

Moreover our keyword detection method suggests that we are concentrating only the words in capital letters. So we have considered the words in all capitals under consideration. It is found that character level accuracy of the selected OCR for those cases in nearly 86.57%. But the accuracy in word level is 78%. The performance of the pre processing module for the set of input images obtained from active pages of a DTH service provider as shown in Figure 4.28 is given in Figure 4.29. From the Figure 4.30 it is evident that considerable improvement is obtained in character recognition after restricting the ROI and applying preprocessing and touching character segmentation before providing the final image to the OCR engine.

Figure 4.28: Input Images for Pre-processing Module

Image	Output of GOCR	Output of Tesseract	After Applying Proposed Algorithms	
			GOCR	Tesseract
(a)	Sta_ring Govind_. Reem_ _n. Rajpal Yadav. Om Puri.	Starring Guvinda, Rcema Sen, Raipal Yadav, Om Puri.	Starring Govind_. Reem_ _n. Rajpal Yadav. Om Puri.	Starring Guvinda. Reema Sen, Raipal Yadav. Om Puri.
(b)	_____ ___ ___ _____ ____ __	Pluww SMS thu fulluwmg (adn In 56633	___ SMS th_ follcmng cod_ to S	Planta SMS tha Iullmmng mda tn 56633
(c)	SmS YR SH to	SMS YR SH in 56633	SmS YR SH to _____	SMS YR SH to 56533
(d)	_m_ BD to _____	SMS BD to 56633	SMS BD to S____	SMS BD to 56633
(e)	AM t___o_,_b _q____	AM 2048eb 141117	AM tOa_gb _q____	AM 2048eb 141117
(f)	_M_= _ _A___ to Sd___	SMS: SC 34393 tn 56533	_M_= _ _A___ to Sd___	SMS: SC34393 tn 56633
(g)	_W _ ' _b _ Ib_lb _a	W6.} 048ab1;lbwzIb1a	___ _Y_b ylbw_Ib_a	WP 2048ab MlbwzIb 1 a
(h)	ADD Ed_J to S____	ADD Eau to $6633	ADD Ed_J to S____	ADD Edu to 56633
(i)	AIC STAIUSIS/OUO_ t_;OS;t_	AIC STATUS25/0209 1 9:05:1 4	mlC S_ATUSIS/OUO _t_;OS=tA	A/C STATUS 25/02/09 1 9:05:14
(j)	_ _____'__	Sub ID 1005681893	WbID_OOS_B_ B__	Sub ID 1005681893

Figure 4.29: Output of the Different OCR Engines before and after Applying the Image Processing Algorithm

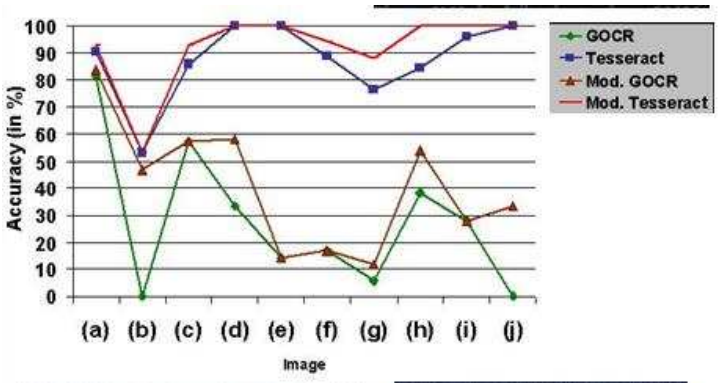

Figure 4.30: Performance of Different OCR Engines Before and After Proposed Image Enhancements

Figure 4.31: Screen Shot of the Google Search Engine With the Recognized Text as Input

Figure 4.32: Screen Shot of the System

4.7.3.3 Accuracy in Required Information Retrieval

The key words are all the words in all capital in the output of OCR. The searching efficiency can be improved by removing the stop words but we have not used it in the system as Google search engine is capable of handling it. Wrong recognition of the OCR module can be overcome by having a strong dictionary or language model. But in the proposed method we get rid of this constraint as the Google search engine itself has one such strong module. So we simply give the output to Google search engine and in turn Google gives the option with actual text as shown in Figure 4.31. We have given the input to Google as "MUMBAI ATTACHED" as it the text detected by the OCR. But Google itself gives the actual text as an option in their "Did you mean" tab. In case of correct recognition the search engine also works properly. One sample screen shot of PC version of the code is depicted in Figure 4.32. In case of erroneous output of OCR, that can not be handled by Google can't return the intended information in the first page. But still the intended information comes in the list box as shown in 4.32.

Chapter 5

Generation of Electronic Program Guide for RF fed connected Televisions

5.1 Introduction

Electronic Program Guide (EPG) is a feature where the program schedule of the channel is shown as a text over the video of the channel user is viewing. The EPG updates the user continuously with scheduling information of current and near future programs that are to be telecasted in the user tuned channel. A recent form of EPG is Interactive Program Guide (IPG). An IPG allows television viewers to navigate scheduling information menus interactively, selecting and discovering programming by time, title, station using an input device such as a keypad, computer keyboard, or TV remote control [140]. Thus EPG and IPG are becoming very popular among the customers of Direct to Home (DTH) Services. Though Digital TV is becoming popular all over the developed countries, the report from Telecom Regulatory Authority of India (TRAI), as shown in Table 5.1, reveals that even in 2008 only 6.55% of the total the number of Houses owning a TV are using Direct to Home (DTH) service and the rest 93.45% are using Radio frequency (RF) Cable as input to the TV. Unlike digital TV transmission it is not possible to automatically get channel information of TV programs from any metadata in cable TV[141]. Therefore in India more than 90% of the TV viewers are unable to view EPG while watching Television (TV).

In some countries, some cable operators dedicate a particular bandwidth to to provide scheduling information about all the channels. This type of EPG is broadcast with a specialized hardware installed at provider's central television distribution facility. But the major challenges behind implementing such a system are as listed below:

- There is no additional information about the future shows of the channel is available as meta data as it happens in case of DTH services.
- Most of the local cable operators are not willing to invest for such an additional hardware

Table 5.1: Survey Report from TRAI, India

Year	No of TV owning home (in Million)	Number of DTH subscriber (in Million)
2006	117	2.3
2007	134	6.5
2008	145	9.5

to show EPG of each channel.

- Even if the EPG is showing in a dedicated channel, it is not convenient for the user to tune to that particular channel every time he/she wants to see the EPG by skipping the program of his/her interest.

- There is no dedicated frequency range where a particular channel is aired. It is very common practice to air the most popular shows in lower frequency range.

So one way to handle the problem is to recognize the channel in the client end of the Home Infotainment Platform (HIP) that can take RF feed cable as input. In this chapter a method to provide an EPG for the viewers of RF feed cable TV is proposed.

Some related work in this field is described in the state of the art section in chapter 1. But it is not possible to use these algorithms for implementing such a system because of the following reasons:

- Usually the number of channel is more than 100. So memory and CPU requirement is quite high for this approach.

Figure 5.1: Logo in Different Location during Different Shows

Figure 5.2: Set Max Logo Color in February 2010 (left) and March 2010 (right)

Figure 5.3: Starplus Logo Till March 2010 (right) and after March 2010 (left)

- Each channel logo needs to be matched with each frame of the video otherwise the channel change can not be identified at the time of change.

- The channel logo of a particular channel is not fixed for all time. One such example is shown in Figure 5.1. In this figure it is shown that the ten sports channel logo is in two different locations during two different shows.

- Some times the content providers changes their channel logo completely or they change the color of the logo. For example the logo of Set Max channel changed their logo color just before Indian Premier League (IPL) in March, 2010. Two logos are shown in Figure 5.2. On the other hand the logo of Star Plus channel changed completely in May 2010. This is depicted in Figure 5.3.

To account for the above problems, we have proposed a system which initially tests whether the channel logo is in its usual location or not. If the logo is not recognized using this method, we try to localize the logo region and then prompt for manual intervention to store the modified logo template or the new channel logo template by adding the new or replacing the existing one.

5.2 System Overview

Overview of the proposed system with related other applications are shown in Figure 5.4. The overall process is as follows:

- Broadcast video is recorded offline using the set top box (like tata sky) and the proposed Home Infotainment Platform (HIP).

- The recorded video is used to generate the template database for all channels. This step involves manual intervention.

- Whenever the video comes to HIP, the channel logo is recognized with the help of channel template stored in database

- This detected channel logo information is used to develop different VAS for Connected TV.

This chapter will mainly focus on the application of providing EPG for connected TV.

The proposed solution is developed on an ARM based dual core platform. The video frame collection and the decision modules are implemented on ARM and the core algorithm for logo score

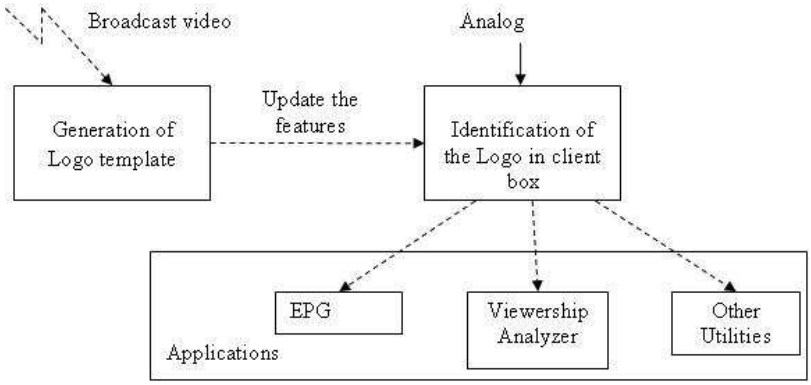

Figure 5.4: Overall System for the Channel Logo Identification

generation is implemented on Digital Signal Processor (DSP). The system environment of the DSP is shown in the Table 5.2.

As the proposed system was developed for providing the EPG for Indian TV channels, initially we have studied the behavior of channel logos for nearly 100 Indian Television channels available in West Bengal. This analysis shows that the channel logos can be classified in 9 different classes as listed below:

- Opaque logo and rectangular shape. List is given in Table 5.2.
- Opaque logo and non rectangular shape. List is given in Table 5.2.
- Transparent Background(BG) and opaque foreground (FG). List is given in Table 5.2.
- Alpha blended BG. List is given in Table 5.2.
- Alpha blended FG and BG. List is given in Table 5.2.
- Static but logo with color changing over time and opaque BG. List is given in Table 5.2.
- Static but logo with color changing over time and transparent BG. List is given in Table 5.2.
- Animated with deterministic pattern.
- Animation with Random motion

Mahua TV, Kairali and NDTV India channels have animated logos with deterministic pattern. The only channel with randomly animated logo is 9xM.

Table 5.2: Hardware Environment for Testing

Parameter	Description
Clock	594 MHz
Internal Memory	32Kbytes L1 program cache, 32Kbytes L1 data cache and 128Kbytes L2 cache
External memory	256Mbytes external RAM
Functional Units	Eight independent functional units of addition and multiply
Internal Registers	Sixty four 32-bit general purpose registers

Table 5.3: List of Channels with Opaque Logo and Rectangular Shape

Sony	DD Bharati	AXN	SAB	Gyan Darshan
CNN	MAX	DD Shayadri	Nick	PIX
DD Bangla	Animax	Sony Aath	Star Gold	MTV
Akash	Star News	Kalianganar T	V 9 Gurati	Star Vijay
IBN Lokmat	Amrita	Star Majha	TV 9 Kanada	PTC News
Star Ananda	NE TV	PTC Punjabi		

Table 5.4: List of Channels With Opaque Logo and Non Rectangular Shape

Zee TV	Ajtak	Zee Cafe	CNBC Awaz	Zee Trendz
CNBC 18	Zee Cinema	Mi Marathi	Z Studio	Nepal 1
Zee News	MH1	Zee Business	OTV	Zee Marathi
Headlines today	Zee Bangla	Hungama	Zee Panjabi	

Table 5.5: List of Channels With Transparent Background and Opaque Foreground

ETV Rajasthan	ETV UP	ETV Bihar	ETV MP	ETV Telegu
ETV 2	ETV Marathi	ETV Kanada	ETV Bangala	ETV Gujrati
ETV Oriya	ETV Urdu	Star One	Sahara One	Astha
Cartoon Network	DD National	Star World	Fashion	National Geo
V	Rajya Sabha	Star Pravah	Show case	Fox History
Jaya TV	DD Chandana	Star Jalsa	Filmy	Animal Planet
Maa TV	DD Malayalam	HBO	Disney	DD Oriya
Suvarna	DD ne	BBC World news		

Table 5.6: List of Channels With Alpha Blended Back Ground

Star Plus	Travel and living

Table 5.7: List of Channels With Alpha Blended FG and BG

Star Utsav	Times Now	XD
Star Movies	Discovery	Pogo

Table 5.8: Static Location but Logo With Color Changing Over Time and Opaque BG

DD News	Loksabha TV	India TV	News 24	NDTV 24x7
CNN	IBN	VH	Raj	TV

Table 5.9: List of Channels Static but Logo With Color Changing Over Time and Transparent BG

Sankskar	DD Urdu	Asia net news	News Live	DD Sports
Isai Arusi	Asia Net	Sakshi	TV 5	Akha

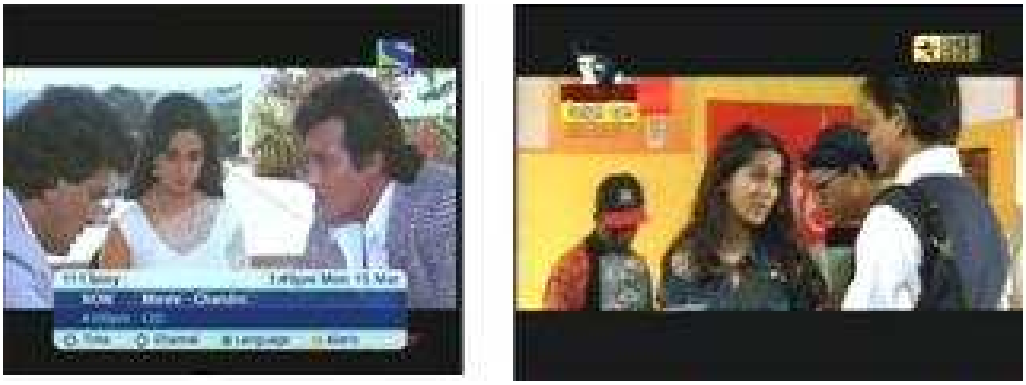

Figure 5.5: Channel Video With a Black Strip in Top and Bottom

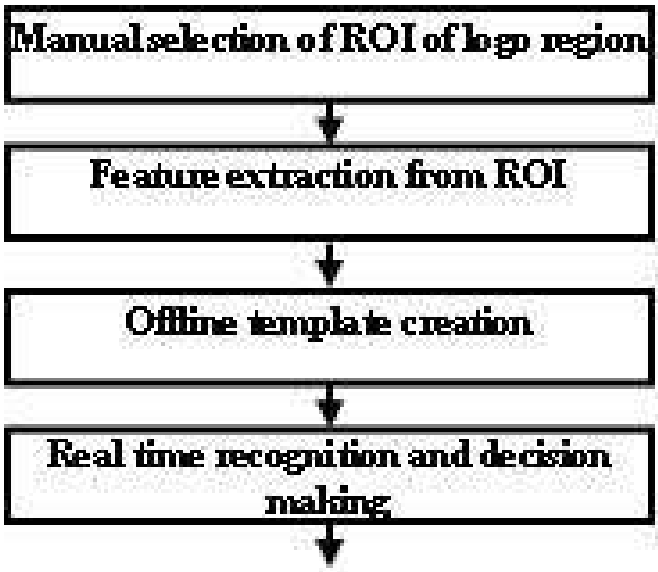

Figure 5.6: Overview of the Algorithm

Our proposed method is based on some observations on the above channels listed below:

- All of the channels under consideration have an unique channel logo
- Channel logo is displayed in any of the four corners in the TV screen i.e. probability to obtain a channel logo in the central part of the screen is zero
- Transparency bits for some pixels in the channel logo are set as 1. So the background color doesn't remain constant through out the video.
- Some videos have a black strip at top and bottom of the video content to make the video content compatible to the TV screen area. One such example is shown in Figure 5.5.

In the proposed solution, initially some instance for each channel is recorded to generate the template offline. To create the template some user intervention is required to mark the logo region. Then the features are extracted from this marked ROI and stored in a template. Finally, this template is used to recognize the channel logo. Overview of the method is described in Figure 5.6. This chapter will mainly present the work published in [142].

5.3 Template generation

The flow diagram of offline template creation is described in the Figure 5.7. The method of offline template creation is as follows:

- Initially list all the channels available and also assign a channel identifier for each of the channels. This channel listing should be exhaustive so that all the channels aired in India is covered. This is referred as "channel info database" in the later part of the chapter.
- Every day record at least one instance of each channel and annotate them manually. This checks whether any channel logo has changed or new channels have been added.
- Store the annotated channel videos as ground truth data base
- Verify whether the channel name in the annotated file is present in the channel info database or not.
- If the channel is not in channel info database, it can be concluded that either it is a new channel or there is a change in the channel logo.

- In case of new channel (i) enter the channel information into channel info data base and (ii) create a template for the channel logo
- In case of change in channel logo generate template for the new channel and update the template database in the Network File System (NFS).
- Initially, when no channel template is there in the template database, all channel logos are treated as a new channel.

5.3.1 Selection of logo region

It is always preferred to select the logo region automatically. But the state of the art on this technology mainly assumes that the logos are static, opaque and have a rectangular shape. Some papers made an attempt to localize the transparent and dynamic channel logos using PCA and ICA to localize the channel logos along with temporal behavior of the entire video frame. But on a DSP platform, that is limited in processing power, it is not feasible to implement such a system. We have implemented a low complexity method for the automatic logo region localization method using the methodology described below.

5.3.1.1 Automatic logo region localization

The method of marking the channel logo region automatically is described below:

- Compute the average frame($a_i(x, y)$) representing the average of the pixel value at (x,y) coordinate from first frame to i^{th} frame iteratively as

$$a_i(x, y) = \frac{\sum_{k=1}^{k=i} f_k(x, y)}{i} \tag{5.1}$$

  where $f_k(x, y)$ represents the pixel value at (x,y) coordinate of the i^{th} frame.

- Compute the dispersion (d_i)of each pixel of each i^{th} frame as

$$d_i(x, y) = d_{i-1}(x, y) + abs(a_i(x, y) - f_i(x, y)) \tag{5.2}$$

- Compute the variation ($v_i(x, y)$) in pixel value at location (x,y) at i^{th} frame as

$$v_i(x, y) = \frac{d_i(x, y)}{i} \tag{5.3}$$

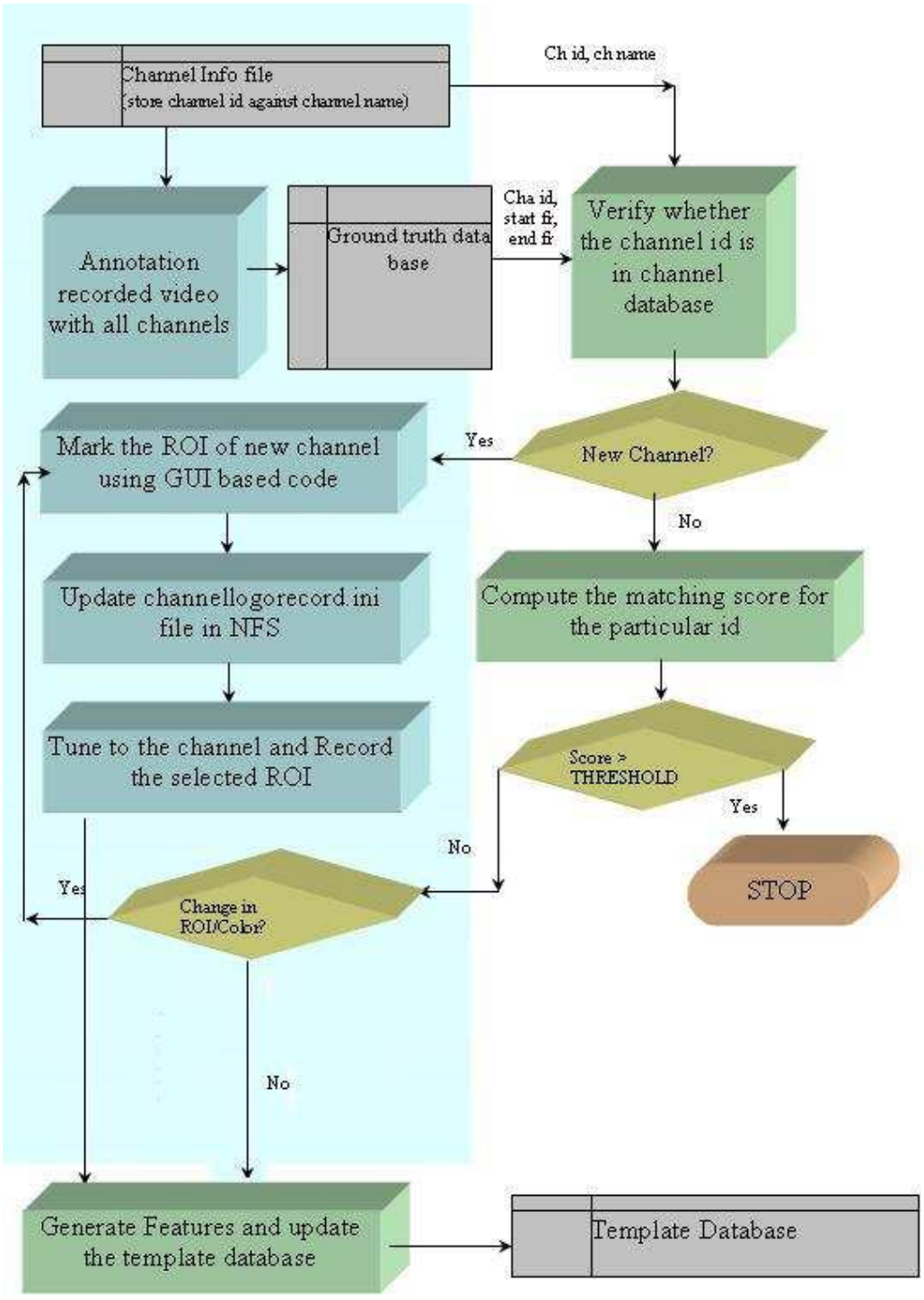

Figure 5.7: Flow Diagram for Offline Template Creation

- Mark the pixels having a variance greater than a empirically obtained threshold Th_{var} as WHITE.

$$f_i(x,y) = WHITE \quad \forall \quad x,y \ni v_i(x,y) > Th_{var} \tag{5.4}$$

- Find connected components for all non WHITE pixels in the frame. Let ($cc_i(j)$) represent the j^{th} component in the i^{th} frame

- Remove all candidate regions not satisfying the size constraint. The size constraint is set based on the heuristic that the candidate logo region must have a size restricted in a particular range of values.

- Mark the remaining components as candidate components ($cand_i(i,j)$) for the i^{th} frame

5.3.1.2 Manual logo region localization

The automatic logo region localization method has the following limitations and thus we have to look for an alternative.

- The computational cost for this localization module on DSP performs is almost 2.5 second per frame. Using the hardware accelerators and intrinsics it could not be reduced below 1.5 seconds per frame

- All the Indian TV channels except 9xM have a single channel logo in a particular region. Even in case of dynamic logos, the channel logo moves within a specific region.

- In case of some channels, the logo resides in two different regions but their positions are fixed.

Therefore we mark the regions manually and store them in a template file. In case the channel logo resides in multiple regions, we store different entries for the same channel. The method of marking the channel logo region manually is described below:

- Initially a video comprising of videos of all possible channels is recorded and annotated manually. The annotated file is stored in a XML file with fields channel code, start frame number and end frame number

- This video is played using a tool that enables the user to select the ROI from the video using a mouse. ROI is the logo region. The tool takes the annotated XML file as input to

generate an output XML file with ROI coordinates, height and width of ROI and features of the ROI

- The feature consists of quantized colors of the pixels in the ROI. We have used quantized colors to reduce the effect of display setting of the TV that incurs a large variation in UYVY format

This section provides the process flow for the steps involved in semi automatic approach for offline template database construction. The approach is depicted in the Figure 5.2. This module is executed every day to check whether any new channel is added or existing channel logos are changed.

- Initially all channels available by the service provider are listed and are stored in a file named "Channel Info". This file needs to be modified once a new channel is added by the service provider. Each data set in this file consists of two fields, channel name and channel id.
- Next, a video consisting of all channels within 5-10 sec duration is recorded.
- The recorded video is manually annotated and stored in a file ("groundtruth.XML") in XML format with following syntax:

  <Ground truth file by manual indexing>

  <Number of channels>

  Number of channels in recorded video

  < /number of channels>

  <Channel id>

  Channel id

  <Start frame>

  Frame number from where that channel is starting

  < /Start frame>

  <End frame>

  Frame number from where that channel is ending

< /end frame>

< /channel id>

< /Ground truth file by manual indexing>

The recorded video, “Channel Info” file, and “groundtruth file” are used to verify whether any insertion or deletion to template database is required.

- The recorded video, “groundtruth.XML” and “Channel Info” file are used as a input to verify whether the channel logo in the video file is already existing in the template file or not.
- If the channel logo information is already present in the template database, the features are extracted from its previously marked Region of Interest (ROI) and matched with the same features in the template data base. If the matching score crosses the threshold, no operation is required. Otherwise it is considered as a new channel logo.
- If the channel logo information is not found in the template database or there is a change in ROI for any logo already present in the template, a semi automatic tool is used to mark the ROI in the video using mouse.
- The video of that particular ROI is recorded for a longer duration (10-15 min).
- Features are extracted from the ROI video and the template data base is updated accordingly.

5.3.2 Data Acquisition

Broadcast TV content is fed into the video capture module via analog tuner or composite video. The video input coming to the hardware is captured using Linux V4L2 interface as analog video input sources. The capture application makes V4L2 ioctl calls to configure the capture driver and de-queue frames from capture array. Once the frames are copied to user space for post processing; the original frame is en-queued back to be overwritten. This is explained in Figure 5.8.

5.3.3 Feature Extraction

Quantized color feature is used for channel logo recognition. The quantized color matrix of the ROI of the annotated video is stored in the template database which in turn is matched with

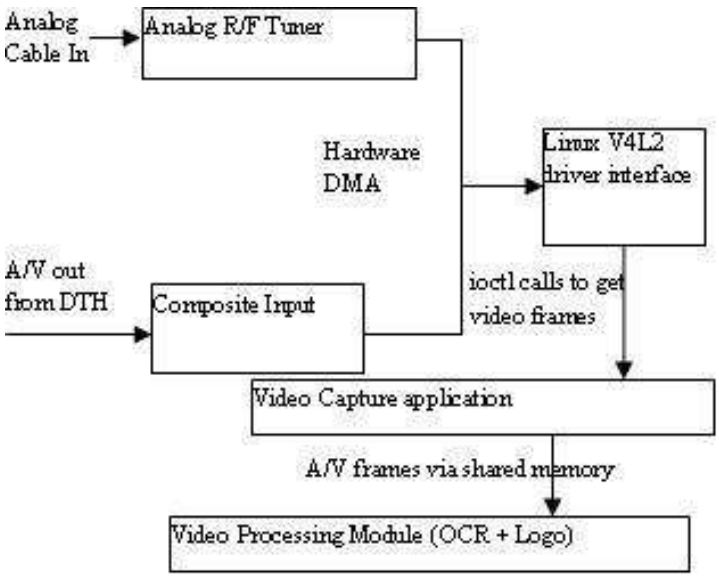

Figure 5.8: Data Acquisition System Overview

the same feature extracted from the ROI of the test video. Among the different classes of channel logo as described earlier in section 5.2, the class 1 logos are easy to detect. But class 2 and class 3 logos do not lead to a good matching score because of the interference occurring due to non static background within the ROI. For class 4, 5 and 6 logos we have used average frame in order to reduce the effect of the background. We have not discussed any method for the 7th class of channel logos. Here are the steps involved in constructing the offline template:

- All the pixels of the ROI are quantized to 36 color beans
- Find the distribution of each pixel in the ROI for 10-15 min of the video. Since we have sampled the video at a frame rate of 25 FPS, the total number of samples for each pixel position would be more than 15000.
- If the standard deviation (STDEV) of pixel is HIGH (i.e. above a pre determined threshold), it suggests that the pixel has high variation over time and thus the pixels of that location is marked as DON'T CARE.
- Store the average pixel value for the pixel positions for which the STDEV is low in the template database along with its ROI information in XML format.

5.3.3.1 Color Quantization

The input for the video format for channel logo region is UYVY. Initially they are converted to HSV color space to separate information of color, intensity and impurities. Initially these HSV values are quantized into 36 beans as described below.

Let h, s, and v be the HSV domain color value, with s and v is normalized between [0,1], for a particular R, G, and B value and index is the quantized bin index. Now, a pure black area can be found in

$$v \in [0, 0.2] \quad and \quad index = 0 \tag{5.5}$$

Gray area can be found in

$$s \in [0, 0.2] \quad and \quad v \in (0.2, 0.8] \quad and \quad index = \lfloor (v - 0.2) * 10 \rfloor + 1 \tag{5.6}$$

So index value will lie in the range of 1 to 6 Similarly, a white area can be found in

$$s \in [0, 0.2] \quad and \quad v \in (0.8, 1.0] \quad and \quad index = 7 \tag{5.7}$$

The color area is found in the range $s \in (0.2, 1.0] and v \in (0.8, 1.0]$, for different h values and index varies in a range of 8 to 35

The quantization of h, s and v value for different s and v is done as following:

Let H_{index}, S_{index}, V_{index} be the quantized index for different h, s and v value.

$$S_{index} = 0, \forall s \in (0.2, 0.5] = 1, \forall s \in (0.5, 1.0] \tag{5.8}$$

$$V_{index} = 0, \forall v \in (0.2, 0.8] = 1, \forall v \in (0.8, 1.0] \tag{5.9}$$

$H_{index} = 0, \forall h \in (330, 360]$ and $h \in (0, 22]$

$= 1, \forall h \in (22, 45]$

$= 2, \forall h \in (45, 70]$

$= 3, \forall h \in (70, 155]$

$= 4, \forall h \in (155, 186]$

$= 5, \forall h \in (186, 278]$

$= 6, \forall h \in (278, 300]$

Finally the histogram bin index is given by the following equation

$$index = 4 * H_{index} + 2 * S_{index} + V_{index} * H_{index} \tag{5.10}$$

5.3.3.2 Pixel of Interest (POI) Detection

As a good number of channel logos are not opaque and not of a rectangular shape, it is necessary to mark the pixel of interests (POI) i.e. the pixels representing the FG part of the logo. The method of marking the POI is described here. The video buffer for the i^{th} frame (f_i) is used to store the quantized color values. The steps involved are as follows:

- Compute the run-time average of each pixel of i^{th} frame at (x,y) coordinate ($a_{x,y}$) as

$$a_i(x, y) = \frac{(a_{i-1}(x, y) * (i - 1) + f_i(x, y))}{i} \tag{5.11}$$

- Compute dispersion ($d_i(x, y)$)of each pixel of i^{th} frame as

$$d_i(x, y) = d_{i-1}(x, y) + abs(a_i(x, y) - f_i(x, y)) \tag{5.12}$$

- Compute variation ($v_i(x, y)$) in pixel value at location (x,y) at i^{th} frame as

$$v_i(x, y) = \frac{d_i}{i} \tag{5.13}$$

- Mark the pixels having a variance greater than threshold as DONT CARE and as POI otherwise.

$$f_i(x,y) = \quad DON'TCARE \quad \forall x,y \in v_i(x,y) > Th_{var} \tag{5.14}$$

5.3.4 Data representation

The XML format in which the template data is stored into template database is shown below:

<Template database>

<Number of channels>

Number of channels in template

< /number of channels>

<Channel id>

Channel id

< /channel id>

<x coordinate of top left position of ROI>

Entry for top left x coordinate

< / x coordinate of top left position of ROI >

< y coordinate of top left position of ROI >

Entry for top left y coordinate

< / y coordinate of top left position of ROI >

< *height* >

$HeightofROI$

< /height>

<width>

Width of ROI

< /width>

<Quantized colors of pixels in ROI or DON'T CARE>

Pixel values or DON'T CARE

< /quantized colors of pixels in ROI or DON'T CARE>

.....Continue for all channels .....

< / Template database >

5.4 Logo recognition

The analog broadcast video is digitized within the HIP and that digitized input video in UYVY format is fed as input for the channel logo recognition module. Features of each candidate region are matched with the features of template logos stored in the template database. In the proposed method a multifactorial approach is used to recognize the channel logo. The factors and the reason behind selecting these factors are as below.

5.4.1 Bhattacharya distance based factor

This feature is used to find the histogram similarity between the candidate and the template. Details about this distance can be found in [143]. Once the quantized coefficients for the hsv values (qhsv) are obtained we obtain the histogram of qshv values. Let h_{cand} and h_{logo} represent the histogram of candidate region and template logo respectively. We obtain the normalized histogram by

$$h_{cand-norm}(i) = \frac{h_{cand}(i)}{pixel - cnt} \quad \forall i \in (1, 2, ..., 36) \tag{5.15}$$

Similarly, we can obtain the normalized histogram for the template logo region as

$$h_{logo-norm}(i) = \frac{h_{logo}(i)}{pixel - cnt} \quad \forall i \in (1, 2, ..., 36) \tag{5.16}$$

where pixel-cnt is the number of pixels in the candidate logo region. Though Bhattacharya's distance function is usually used to compute the distance between two probability distributions, we have used it to compute the difference between two histograms. We assign the factor value (f_{bhat}) for this feature as,

$$f_{bhat} = 1 - \sqrt{1 - \sum_{i=1}^{i=36} \sqrt{h_{logo-norm}(i) * h_{cand-norm}(i)}} \tag{5.17}$$

Now, as the Bhattacharya distance between two normalized histogram lies in the range of 0 to 1, the factor value lies in the range of 0 to 1. But as the factor value indicates the membership value, we subtract the Bhattacharya's distance from 1 so that the best match, for which distance is 0, is represented by the factor value 1.

5.4.2 Crossing count and run length similarity based factor

These two shape invariant features are used in different pattern recognition problems like Optical Character Recognition (OCR). We use these features in the following manner.

Construct Similarity matrix: We define similarity matrix as the matrix of size 36x36. It stores the number of occurrence of each quantized HSV (qhsv) transition in the form of an adjacency matrix.

$\forall y \in (0, height)$

$\forall x \in (0, width - 1)$

$i \leftarrow [y][x]$

$j \leftarrow [y][x + 1]$

$cross - mat[i][j] \leftarrow cross - mat[i][j] + 1$

$cross - mat[i][j] \leftarrow \frac{cross - mat[i][j]}{height * (width - 1)}$

Now $cross - mat[i][j]$ stores the normalized values only. Next, we compute the same normalized crossing count matrix from the logo template. Let that be denoted as $cross - mat - ref[i][j]$. Finally the crossing count based factor (f_{cross})is defined as

$$f_{cross} = abs(cross - mat[i][j] - cross - mat - ref[i][j]) \forall i \in (0, height) \quad and \quad j \in (0, width) \quad (5.18)$$

Now f_{cross} will also lie in the range of 0 to 1.

5.4.3 Aspect Ratio based factor

From our observation of all the channel logos it is found that the aspect ratio (width/height ratio) of the logo for a particular family of channels is almost the same and it is quite different from other set of logos. For example the aspect ration for all Z series of channel (Zee Studio, Zee Business etc.) is same but it is quite different from the Sony set of channels (like Sony, Set max, Sab). Thus this feature works effectively as a first level classifier. It is a good metric in segregating between different family of channels.

Let the aspect ratio of the candidate region be asp_{cand} and the aspect ratio of the channel logo in the corresponding template be asp_{logo} , then the aspect ratio factor $f_{asp \quad i}$ indicating the similarity score with ith channel logo is represented by

$$f_{asp \quad i} = 1 - \frac{abs(asp_{cand} - asp_{logo})}{Max_k(abs(asp_{cand} - asp_{logo}))} \forall k \in (1, n) \quad (5.19)$$

where n is the number of channel logo available in the template. This feature also returns the factor value asp_{cand} in a range of 0 to 1 where 1 indicates the best match.

5.4.4 Color correlation based factor

This factor is used where the histogram is similar but the images are distinct in nature. Here we simply compute the correlation coefficient of the POI of the template and the candidate frames. The method is as follows:

- Find the mean (μ_{logo})and standard deviation (σ_{logo}) of all pixels in the POI in the template
- Find the mean (μ_{cand})and standard deviation (σ_{cand}) of all pixels in the POI in the candidate frame
- Subtract the mean from all pixels in the POI in the template
- Perform the same operation for all the pixels in POI for the candidate frame under inspection

Now compute the correlation based factor score (f_{corr}) as

$$f_{corr} = \frac{\sum_{i=0}^{i=n} (p_{cand}(i) - \mu_{cand}) * (p_{logo}(i) - \mu_{logo})}{\sigma_{logo} * \sigma_{cand}} \tag{5.20}$$

where n is the number of pixels in POI. As the normalized correlation coefficient score lies in the interval of (0,1), the factor value also lies in the same range.

5.4.5 Construction of evaluation matrix

The evaluation matrix is formed using the factor values as described below

$$V = \begin{bmatrix} f_{bhat1} & f_{bhat2} & \cdots & f_{bhatm} \\ f_{asp1} & f_{asp2} & \cdots & f_{aspm} \\ \vdots & \vdots & \cdots & \vdots \\ f_{corr1} & f_{corr2} & \cdots & f_{corrm} \end{bmatrix} \tag{5.21}$$

5.4.6 Construction of Additive Standard Multifactorial (ASM) function

Now, from the evaluation matrix V, it is difficult to obtain any solution to the decision-making problem. So we define a mapping function M_m that maps the m-dimensional vector $f = (f_1, f_2, \cdots\cdots f_m)$ into a one dimensional scalar i.e. $M_m f = M_m(f_1, f_2, \cdots\cdots f_m)$ we apply ASM on V to obtain the multifactorial evaluation $V' = (v_1, v_2, \cdots\cdots v_n)$ where $v_i = M_m(v_{1i}, v_{2i}, \cdots\cdots v_{ni}) \quad \forall i =$

$[1, n]$ Now we define a mapping function M_m as a simple arithmetic average over the m number of factors. So we finally obtain the decision making matrix V' which is a row matrix and each element represents the confidence score of membership of the candidate region matching with the i^{th} logo. The score always gives a value in the range of 0 to 1. We consider the logo as a candidate if the score is greater than 0.75.

5.4.6.1 Logo Recognition

Due to noise and dynamically changing conditions, the algorithm suffers from either too many false positives (if the threshold is kept low) or detection misses (if the threshold is kept too high). To resolve this problem, we have adopted the standard M/N detection approach used in Radar Detection theory. The scores are accumulated for N consecutive frames. The channel id that is occurring at least M times is detected as the recognized channel. We have taken M as 5 and N as 9.

- First take four corner areas as specified in assumption 1.
- Quantize each pixel of those four regions using the method described in section 5.2.1
- Match the test video frame with each of the channels in the template
- Find the Correlation Coefficient (CC) for each pixel in POI of template
- Find the channel id for which CC is maximum
- If CC is greater than 0.75, mark it as a candidate match
- Accumulate the result of 9 such consecutive frames
- Use cipher method and M/N method to conclude about the detected channel

5.4.7 Detection of channel change event

This module identifies any change in channel by detecting the blue screen that comes during channel transitions. The state transition for the logo recognition process is shown in Figure 5.9. In the proposed system, the logo recognition is performed every N (15) seconds or after a channel change.

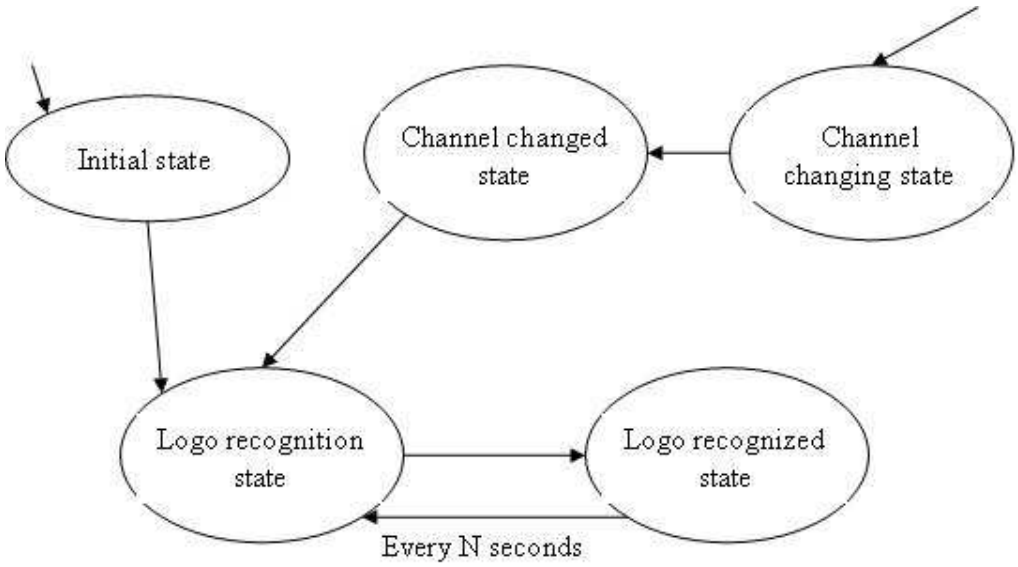

Figure 5.9: State Transition for the Logo Recognition Process

5.5 EPG generation and Rendering

5.5.1 EPG generation

The proposed system is connected to the Internet through wireless broadband service. A boot up, the system will connect to Internet automatically (or on user intervention) and will download all the EPG content for all the available channels from some well defined dependable source. The EPGs for most of the Indian TV channels are available from www.whatsonindia.com. Theis data of each channel is stored in XML file format. For example, system can store program details for DD News channel in a file 4.XML where 4 is the predefined channel id for DD News. Structure of the XML file would be like this:

< Channel >

<ChName>DD News< /ChName>

<Events>

<Entry>

<Key>News< /Key>

<Date>15022010< /Date>

<sTime>2:00 PM< /sTime>

<cTime>2:30 PM< /cTime>

<Name>Samachar< /Name>

<Dir>

< /Dir>

<Cast>

< /Cast>

<Votes>0< /Votes>

<Fav>0< /Fav>

$< ULik > NO < /ULik >$

<UFav>NO< /UFav>

<Syn>

The most recent news, updates and developments

< /Syn>

< /Entry>

Figure 5.10: Small Programme Info View

<Entry>

<Key>News< /Key>

<Date>15022010< /Date>

<sTime>2:30 PM< /sTime>

$< cTime > 3:00PM < /cTime >$

<Name>Mid Day News< /Name>

<Dir> < /Dir>

<Cast>

< /Cast>

<Votes>0< /Votes>

<Fav>0< /Fav>

<ULik>NO< /ULik>

<UFav>NO< /UFav>

<Syn> The most recent news, updates and developments < /Syn>

< /Entry>

< /Events>

< /Channel>

Following steps show how we parse XML files to obtain program related data for a particular

Figure 5.11: More Info View

Figure 5.12: Full EPG View.

channel.

- Detect the channel logo and use a map table to get the channel name (or a predefined channel id).
- Create some dynamic array like $ProgType[]$, $Name[]$, $Duration[]$, $ProgType[]$, $Story[]$ etc.
- Loop through the file and store related data in corresponding arrays whenever we encounter tags like $< Key >$, $< Date >$, $< Name >$, $< sTime >$, $< Syn >$ etc.
- Save the formatted data for rendering

5.5.2 Rendering

These data can be rendered over the TV show in different mode. When a person switches to a new channel, we detect the channel logo, parse the XML file and match the time frame to show the user a short information box containing only program name, duration, channel name. This comes up automatically when a user changes channel using his/her TV remote. It tells the user what is presently on air, as shown in Figure 5.10.

In a second mode, as shown in Figure 5.11, if a user presses a "more info" button from the remote, the system pops up the details of the program containing all available info for it. This is a user interrupted mode of showing details, because it takes up more screen space than the previous view mode and will block video content. Therefore, it should ordinarily be hidden and visible only on user demand.

A third mode can be added on to the first mode. User, on pressing a remote button, can see a short info about the programs presently on air and what is coming up next. A fourth mode as shown in Figure 5.12 is viewing a full EPG which will show the entire program list for all the channels for the whole day or perhaps the whole week. This will be a fully navigable tabular view along with TV being shown in the upper right corner of the screen. Figure 5.12 depicts a full EPG view. User can navigate through this EPG table both horizontally and vertically. On focusing on a particular program on this EPG table, user can see all the details of this program on upper left corner of the TV screen.

5.6 Experimental results and discussion

5.6.1 Testing Environment

The proposed method is tested in both platforms, namely x86 based PC and the above mentioned DSP platform. The main reason behind is that it is not possible to test the accuracy of the recognition module against the stream video without manual intervention. On the other hand the performance in terms of response time can not be evaluated without DSP environment. So we have tested the proposed system against two parameters namely (i) accuracy and (ii) time of response in two different platforms. We have recorded all the channels and annotated it frame by frame. This recording also includes the channel change scenario. Then we use this annotated XML file to test the accuracy of the algorithm. We have also noted down the time to recognize the channel logo and render the EPG on DSP platform.

5.6.2 Method of Testing

In this section we shall describe the testing method for both the parameters.

5.6.3 Recognition Accuracy

Accuracy of the proposed method is tested as below:

- The solution runs on the target DSP platform
- User is asked to change channel manually and note whether the pop up showing the recognized channel name is correct or not. We have applied an automatic method for testing the recognition accuracy, too. This automatic method is as below:
  - Record the streamed video with number of channels and this channel change instances are marked manually.
  - This annotation is stored in a XML file
- Then the recorded video is manually annotated and stored in a file ("groundtruth.XML") in XML format as described earlier:
- The recorded video, and "groundtruth file" is used to test the accuracy of the algorithm.
- The misses and the false positives are recorded as a report.

The accuracy of the proposed method is described against two parameters namely Recall and Precision. Recall (r) and Precision (p)can be defined as:

$$r = \frac{c}{c+m} \tag{5.22}$$

and

$$p = \frac{c}{c+fp} \tag{5.23}$$

where c is the number of correct recognition, m is the number of misses and fp is the number of false positives. The channel logo recognition module is tested over 80 Indian channels and the experimental results reveals that the recall rate is .97 and precision is .99. Our algorithm is developed for providing EPG as a VAS. So we have studied the user behavior as a part of requirement gathering. It reveals that users prefer to see the EPG within 1 second of tuning to a particular channel. In case of rejection, the user can wait upto 5 seconds to get the EPG but never wants to see the EPG of some other channel. For example, if a user is tuned to HBO and the proposed method can not recognize the channel logo. In this case we prefer not to display any EPG until the recognition module recognizes the channel with enough confidence. It is not allowed to show the EPG of some other channel, say Discovery instead of HBO. We have tuned the Threshold values so that the precision is maximized even it sacrifices the recall value. Still we encounter misclassification that results in a precision value below 1.

The reason behind false negatives is as follows:

- The channel logos with very small number of pixels representing the foreground pixel of the channel logo are missed in 1% cases.
- The reason behind the misses for rest 2% cases is that the channel logo is shifted to a different location from its usual position.
- Some times the channel logo color or the channel logo itself gets changed. In such cases the proposed method can not recognize the channel logo unless the new template and the new location information are added in the template database.

The confusion matrix of the algorithm is shown in the Table 5.10. From the confusion matrix it is evident that most of the channels are mainly confused with DD Ne. The major reason behind it is that DD Ne channel logo has a very small number of POI. So whenever we are computing

Figure 5.13: Advertisement of One Channel in Another Channel

co-relation coefficient, any channel logo that has a also less number of POI and very similar color to DD Ne logo, gives a very high matching score.

5.6.4 Response Time

The performance in terms of response time can be measured in terms of time to recognize the channel logo, parse the XML file to obtain the EPG and render it. These time requirements are shown in Table 5.12. All times in this table are measured in seconds. We have quoted the figures for 13 channels only just to give the essence of expected time requirement.

5.7 Discussions

Thus, in this chapter we have described a novel method to provide EPG as a VAS for connected TV even if connected through an RF cable. The proposed method also describes a method for channel logo localization. If there is enough resource in the target hardware then that localization method can be deployed as well. Here, we have implemented the prototype in a PC based environment and then ported it to a commercially available DSP platform where the proposed algorithm works in real time with same accuracy. Thus, the proposed solution can provide the EPG for the most of TV viewers in India. Moreover the proposed solution can be extended to generate a real time analysis of TV viewership if the data can be sent to a server using a return path. The

Table 5.10: Confusion Matrix for Channel Logo Recognition

Channel Name	Detected as
Zee Trendz	DD Ne/Amrita
Zee Punjabi	TV9 Gujrati
DD News	DD Ne
Nick	No Channel/DD Ne
Nepal 1	Zee Cinema

EPG framework in our proposed system can be moved to a server so that on demand EPG can be implemented to reduce stress on a local system. The system can also pull down EPG related data from internet and mash them up before showing. These data can further be used for different applications such as the rating of a film, review of a program, recommendation on same genre etc. Also chat level recommendation to friends is possible while watching TV. Also user can SMS what he/she is watching now to others who are not in front of TV or watching other channel. The current work can be improved by considering some system level aspects. Currently the channel recognizer module is called once in a second as it can not perform in 30 frames per second on the target DSP platform. The proposed method can also be improved by having a temporal filtering over the recognized channel ID to reduce the scope of false positive.

Table 5.11: Optimization Using Hardware Accelerators for Channel Logo Recognition

Module	Before	After	% saving
IMG-CORR-gen	6.601 ms	5.89 ms	10.77%

Table 5.12: Performance of Proposed System : Time Complexity

Channel	Time to detect channel *	Time to parse EPG *	Time to render EPG *
Star One	1.95	0.45	0.5
Zee Business	1.8	0.4	0.6
Sony Entertainment TV	1.95	0.45	0.5
AXN	1.9	0.5	0.5
Channel V	1.85	0.5	0.5
POGO	1.9	0.3	0.6
Zee TV	1.8	0.4	0.5
Star Movies	1.8	0.4	0.55
Aastha	1.85	0.4	0.5
MH1	1.8	0.5	0.5
Star News	1.9	0.3	0.55
Zee Trendz	1.8	0.4	0.5
ETV Bangla	1.95	0.4	0.6

Chapter 6

Conclusion

The motivation behind the present thesis was to provide some value added services (**VAS**) for an already developed connected TV or interactive set top box. Embedded realization of an efficient video codec, real time video security for the same target DSP platform, text and object recognition from video are the major tasks to accomplish such Value added services. The following goals were set up.

6.0.1 Goals

The thesis had set the following goals:

- *Realization of H.264 Video encoder on target DSP platform*: A H.264 video encoder needs to run in at least 5 Frames per second for QCIF (176x144) resolution on low cost DSP platform. The video encoder should be optimized in algorithm level as well as programming language level to achieve platform independent optimization. Moreover it is required to optimize using the intrinsic and assembly language which is specific to that target hardware.

- *Finding a suitable video encryption methods applicable to H.264 video*: A robust video encryption method compatible to H.264 encoder needs to be implemented to ensure the security of the video content for connected TV with PVR. Moreover the algorithm should be computationally less expensive so that it can run on the target hardware.

- *Robust and imperceptible video watermarking solution*: There is a need for inserting watermark into the video for copyright protection. The watermarking scheme should not degrade the video quality, too. Moreover the watermark insertion and detection should run in real time in the target hardware.

- *Recognition of text regions from video*: A lot of value added services can be proposed by recognizing the context from the texts in the video. But one major challenge behind this set of problem is to localize the text regions from the video. Some algorithms for localizing the text regions from document images are there. But these algorithms cannot perform well in case of low resolution and complex background of videos. Also they are computationally very expensive and thus difficult to implement on a DSP platform.

- *Identifying the breaking news from news video*: Once the text regions are identified from the video, another major challenge is to identify the important part from those texts. In a news video, some texts come as ticker text, some texts are detailed news and some texts are representing the breaking news. The Breaking news are always the news of attraction for that instance of time. So it is quite useful to identify the breaking news from the news video.

- *Providing EPG for RF cable feed TVs*: Website of the TV channels always contains the information about the show in their channel. They always have the information about genre of the show, actor, actress and some description about the show. But in case of RF feed cable TV, it is difficult to recognize the channel, an user is viewing. It is also difficult to parse the EPG content from the website and render it in real time as a blended text over the video.

6.0.2 Goals Achieved

- A method for implementing H.264 video on target DSP platform has been proposed. This method initially includes an study on the complexity of different modules of H.264 encoder. This analysis is very useful for future reference. Moreover both the platform specific and platform independent optimizations for H.264 video encoder have been proposed.

- A two fold video encryption technique has been proposed in the thesis. The proposed method meets the criteria of robustness and real time performance criteria. The proposed method uses the hardware identification number of each box as the key for encryption and thus the key is unique for all the boxes.

- An watermarking method for H.264 video has been proposed. The watermarking method is robust, imperceptible, and blind. This method can also ensure the integrity. Moreover a method for evaluating the watermark is also proposed in the thesis. This evaluation tool can be used in future researcher endeavor to evaluate newly proposed watermarking scheme.

- A text region localization method is described. The proposed methods can extract the text regions when (i) the raw video or (ii) the compressed video comes as input. The method is tuned so that it may have some false recognition but no rejections. Proposed method can extract text from a video frame and also is capable of filtering out the noises and segment the touching characters.

- A heuristic method for spotting the breaking news from the video text has been proposed. This method can identify only the breaking news from the video and can thus form an efficient string as query for the web based search engine.

- A method for channel logo recognition has been proposed. This method can take video input from analog RF feed cable or from DTH and recognizes the channel logo within it. Moreover a method to render the EPG is also described in this thesis.

6.0.3 Scope of Future Research

The study presented here can be extended in several directions. Some of them are highlighted below:

- *Optimization of the H.264 Video Encoder*: The proposed method describes the optimization for most of the computationally expensive units for baseline profile only. This work can be further extended for platform independent optimizations for the other computationally expensive modules like CABAC, Bi directional prediction. Moreover the platform specific optimization for other common platforms like Trimedia can be a scope of future research.

- *Video Screen Layout Segmentation*: The layout of a video is very complex. We have tried to run different document page layout segmentation techniques on different video frames

of news video. But none of these methods can produce a significant result.

- *Frame by frame annotation of video frame using multimodal cues*: The proposed method for mash up of web information and TV context is based on textual content of video only. But a better result perhaps can perhaps be obtained if multimodal cues like audio and image can be used. This can be used for annotating the frames and indexing the video.

- *Cross lingual Information Retrieval*: The textual content from the news video can be further used to retrieve related information from other languages. Script identification and subsequently Cross lingual Information Retrieval (CLIR) are further research issues involved with this problem.

- *Automatic channel logo region identification*: We have found that the channel logo region identification for the animated channels is a challenge. Automatic localization for these channels (like 9xM) is a possible future extension of the present research.

Bibliography

[1] A. Wooldridge, "A special report on innovation in emerging markets", *The Economist*, Page(s) 6, 17th April, 2010.

[2] B. Stelter, "A TV-Internet Marriage Awaits Blessings of All Parties", *http://www.nytimes.com/2011/01/10/business/media/10tv.html?_r* = 2, 9th January, 2011. Last accessed on 14th January, 2011.

[3] A. Pal, C. Bhaumik, M. Prashant, A. Ghose, "Home Infotainment Platform", *Proc. of International Conf. on Ubiquitous Computing and Multimedia Applications, (UCMA2010)*, Miyazaki, Japan, June 2010.

[4] Economic Times, "Dialog can raise internet penetration", *Economic Times Kolkata*, Section-Business and IT, Page(s)5, Apr 27, 2010.

[5] "SMART launches SurfTV", *http://smart.com.ph/corporate/newsroom/SurfTV.htm*, Last accessed on 13th Jan, 2011.

[6] "World Internet Usage Statistics News and World Population Stats", *http://www.internetworldstats.com/stats.htm.*, Last Accessed on Oct 2010.

[7] "ITU report baffled over RP's high mobile phone, TV penetration standard document styles", *http://technews.com.ph/?p=1627*, Last Accessed on Oct 2010.

[8] W. Cooper, "The interactive television user experience so far", In *Proceeding of the 1st international conference on Designing interactive user experiences for TV and video*, October 22-24, 2008, Silicon Valley, California, USA.

[9] A. Pal, M. Prashant, A. Ghose, and C. Bhaumik, "Home Infotainment Platform - A Ubiquitous Access Device for Masses", In *Ubiquitous Computing and Multimedia Applications*, Springer, Volume 75, 2010, Page(s) 11-19, DOI: 10.1007/978-3-642-13467-8, ISBN 978-3-642-13466-1.

[10] T. Chattopadhyay and C. Agnuru, "Generation of Electronic Program Guide for RF fed TV Channels by Recognizing the Channel Logo using Fuzzy Multifactor Analysis", *Proc. of International Symposium on Consumer Electronics (ISCE'10)*,7-10 June, Germany, 2010.

[11] I. E. G. Richardson, "H.264 and MPEG-4 Video Compression", *ISBN 0-470-84837-5*, 2003.

[12] X. Kim and C. C. Jay Kuo, "A Feature-based Approach to Fast H.264 Intra/Inter Mode Decision", *In Proc. of IEEE Int. Symp. Circuits and Systems (ISCAS'05)*, Page(s)308-311, May 23-26, 2005.

[13] H. Wang and Z. Zhu, "Fast Mode Decision and Reduction of the Reference Frames for H.264 Encoder", *In Proc. of Int. Conf. On Control and Automation (ICCA'05)*, Vol. 6, Page(s) 1040-1043 , June 2005.

[14] B. Feng, G. Zhu, and W. Liu, "Fast Adaptive Inter Mode Decision Method for P Slices in H.264", *In Proc. of IEEE 3rd Int. Conf. on Consumer Communications and Networking (CCNC'06)*, Page(s) 745-748, 2006.

[15] Y. V. Ivanov, and C. J. Bleakley, "Survey and Pareto Analysis Method for Coding Efficiency Assessment of Low Complexity H.264 Algorithms", *In Proc. of 10th Irish Machine Vision and Image Processing Conf. (IMVIP) (2006)*, Page(s) 172-179, 2006.

[16] Y. V. Ivanov and C. J. Bleakley, "Skip Prediction and Early Termination for Fast Mode Decision in H.264/AVC", *In Proc. of Int. Conf. on Digital Communications (ICDT)*, August, 2006.

[17] C. S. Kannangara, I. E.G. Richardson, M. Bystrom, J. R.Solera, Y. Zhao, A. MacLennan, and R. Cooney, "Low- Complexity Skip Prediction for H.264 Through Lagrangian 969 Cost Estimation", *IEEE Trans. Circuits Syst. Video Technology*, Vol 16, No 2, Page(s)202-208, 2006.

[18] Y. Kim, Y. Choe, and Y. Choi, "Fast Mode Decision Algorithm using AZCB Prediction", *In Proc. of Int. Conf. on Consumer Electronics. (ICCE'06),Digest of technical papers.*, Page(s)33-34, Jan. 7-11, 2006.

[19] G. L. Li, M. J. Chen, H. J. Li, and C. T. Hsu, "Efficient Motion Search and Mode Prediction Algorithms for Motion Estimation in H.264/AVC", *In Proc. of IEEE Int. Symp. Circuits and Systems (ISCAS'05)*, Page(s)5481 - 5484, May 23-26, 2005.

[20] Y. V. Ivanov, and C. J. Bleakley, "Dynamic complexity scaling for real-time H.264/AVC video encoding", *Proceedings of the 15th international conference on Multimedia.*, Page(s) 962-970, 2007.

[21] H. C. Lin, Y. J. Wang, K. T. Cheng, S. Y. Yeh, W. N. Chen, C. Y. Tsai, T. S. Chang, and H. M. Hang, "Algorithms and DSP implementation of H.264/AVC", *Asia and South Pacific Conference on Design Automation, 2006*, 24-27 Jan. 2006.

[22] S. Kant, U. Mithun, and P. S. S. B. K Gupta, "Real time H.264 video encoder implementation on a programmable DSP processor for videophone applications", *International Conference on Consumer Electronics, Jan 2006*, Page(s) 93-94, 2006.

[23] T. Chttopadhyay, S. Banerejee, and A. Pal, "Enhancements of H.264 Encoder performance for video conferencing and videophone applications in TMS320C55X", *In Proc. of the 10th International Symposium on Consumer Electronics (ISCE'06)*, Page(s) 213-218, Russia, 2006.

[24] G. N. Rao, R. S. V. Prasad, D. J. Chandra, and S. Narayanan, "Real-Time Software Implementation of H.264 Baseline Profile Video Encoder for Mobile and Handheld Devices Acoustics", *In Proc. of IEEE Int. Conf. on Speech and Signal Processing (ICASSP'06)*, Vol 5, Page(s)457-460, May 14-19, 2006.

[25] Z. Wei and C.i Cai, "Realization and optimization of DSP based H.264 encoder", *In Proc. of IEEE International Symposium on Circuits and Systems, 2006. ISCAS 2006*, 2006.

[26] K. W. Hsu, L. Xiang, and R. Chopra, "An IC design for realtime motion estimation for H.264 digital video", *In Proc. of 48th Symp. On Circuits and Syst.*, Page(s) 1489-1493, August 7-10, 2005.

[27] T. C. Chen, C. J. Lian, and L. G. Chen, "Hardware architecture design of an H.264/AVC video codec", *In Proc. of the 2006 conference on Asia South Pacific design automation (ASP-DAC '06)*, January 2006.

[28] S. Y. Shih, C. R. Chang, and Y. L. Lin, "A near optimal deblocking filter for H.264 advanced video coding", *Asia and South Pacific Conference on Design Automation, 2006.*, Page(s) 24-27, 2006.

[29] T. Chttopadhyay and A. Pal, "A Survey on Video Security with Focus on H.264: Steganography, cryptography and watermarking techniques", *Proc. of the 2nd National Conference on Recent Trends in Information System (ReTIS 2008)*, Page(s) 63-67, Kolkata, India, 2008.

[30] S. Bhattacharya, T. Chttopadhyay, and A. Pal, "A Survey on Different Video Watermarking Techniques and Comparative Analysis with Reference to H.264/AVC", *Proc. of 10th International Symposium on Consumer Electronics (ISCE'06)*, Page(s) 616-621, Russia, 2006.

[31] Y. Ye, X. Zhengquan, and L. Wei, "A Compressed Video Encryption Approach Based on Spatial Shuffling", *Proc. of 8th International Conference on Signal Processing*, Volume 4, Page(s)16-20, Greece, 2006.

[32] Y. Li, L. Liang, Z. Su, and J. Jiang, "A New Video Encryption Algorithm for H.264", *Proc. of Fifth International Conference on Information, Communications and Signal Processing (ICICS'05)*, Page(s) 1121-1124, Thailand 2005.

[33] Y. Zou, T. Huang, W.Gao, and L. Huo, "H.264 video encryption scheme adaptive to DRM", *IEEE Transactions on Consumer Electronics*, Vol.52, no.4, Page(s) 1289-1297, Nov. 2006.

[34] S. Changgui and B. Bhargava, "An efficient MPEG video encryption algorithm", *Proc. of Seventeenth IEEE Symposium on Reliable Distributed Systems*, Page(s)381-386, 20-23 Oct 1998.

[35] L. Qiao and K. Nahrstedt, "Comparison of MPEG encryption algorithms", *International Journal on Computers & Graphics, Special Issue: "Data Security in Image Communication and Network"*, Vol. 22, No. 3, Page(s) 437-448, 1998.

[36] A. S. Tosun and W. C. Feng, "Efficient multi-layer coding and encryption of MPEG video streams", *IEEE International Conference on Multimedia and Expo, 2000. ICME 2000*, Vol. 1, Page(s): 119 - 122, 30 July-2 Aug. 2000.

[37] Z. Liu and X. L. Sch, "Motion vector encryption in multimedia streaming", *Proc. of 10th International Multimedia Modelling Conference*, Page(s) 64-71, Australia, 2004.

[38] F. Hartung and B. Girod, "Digital watermarking of raw and compressed video", *in Proc. SPIE Digital Compression Technologies and Systems for Video Commun.*, vol. 2952, Oct. 1996.

[39] F. Hartung and B. Girod, "Fast public-key watermarking of compressed video", *in Proc. IEEE Int. Conf. on Image Processing 1997 (ICIP '97)*, Page(s) 528-531, Santa Barbara, CA, Oct. 1997.

[40] I. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for images, audio and video", *in Proc. IEEE Int. Conf. Image Processing (ICIP 96)*, Lausanne, Switzerland, Sept. 1996.

[41] I. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Digital watermarking of uncompressed and compressed video", *Signal Processing (Special Issue on Copyright Protection and Access Control for Multimedia Services)*, vol. 66, no. 3, Page(s). 283-301, 1998..

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

[43] C. T. Hzu and J. L. Wu. "Digital watermarking for video", *13th International Conference on Digital Signal Processing Proceedings, 1997. DSP 97*, Page(s) 217-220, Greece, July 1997,

[44] C. T. Hsu, "Digital watermarking for images and videos", *Ph.D. dissertation*, Commun. Multimedia Lab., National Taiwan Univ, 1997.

[45] C. T. Hsu and J. L. Wu, "Hidden signatures in images", *in Proc. IEEE Int. Conf. Image Processing (ICIP 96)*, Page(s) 223-226, Lausanne, Switzerland, Sept. 1996.

[46] G. C. Langelaar, R. L. Lagendijk, and J. Biemond, "Realtime labeling methods for MPEG compressed video", *in Proc. 18th Symp. Information Theory*, The Netherlands, May 1997.

[47] R. Tu, L. Zhu, and J. Zhao, "An Adaptive Differential Energy Watermarking Algorithm for Video Signal", *The 32nd Biennial Symposium on Communications*, Page(s) 92-94, June 1-3, Canada 2004.

[48] M. Swanson, B. Zhu, and A. Tewfik, "Multiresolution video watermarking using perceptual models and scene segmentation", *in Proc. IEEE Int. Conf. Image Processing 1997 (ICIP '97)*, vol. 2, Page(s) 558-561, Santa Barbara, CA, Oct. 1997.

[49] M. Swanson, B. Zhu, and A. H. Tewfik, "Multiresolution scenebased video watermarking using perceptual models", *IEEE J. Select. Areas Commun. (Special Issue on Copyright and Privacy Protection)*, vol. 16, pp. 540-550, May 1998.

[50] J. P. Linnartz, "MPEG PTY marking", *WWW: http://diva.eecs.berkeley.edu/ linnartz/ pty.html*, alst accessed Dec, 2010.

[51] V. Darmstaedter, J. F. Delaigle, D. Nicholson, and B. Macq, "A block based watermarking technique for MPEG-2 signals: Optimization and validation on real digital TV distribution links", *in Proc. European Conf. Multimedia Applications, Services, and Techniques-ECMAST '98*, Berlin, Germany, May 1998.

[52] J. Dittmann, M. Stabenau, and R. Steinmetz, "Robust MPEG video watermarking technologies", *in Proc. ACM Multimedia '98*, Bristol, U.K., Sept. 1998.

[53] F. Deguillaume, G. Csurka, J. O. Ruanaidh, and T. Pun, "Robust 3D DFT video watermarking", *Proceeding of the SPIE*, Vol. 3657, page(s) 113-124.

[54] C. Busch, W. Funk, and S. Wolthusen, "Digital watermarking: From concepts to real-time video applications", *IEEE Comput. Graphics Applicat*, Page(s) 25-35, Jan. 1999.

[55] T. Kalker, G. Depovere, J. Haitsma, and M. Maes, "A video watermarking system for broadcast monitoring", *in Proc. SPIE IS&T/SPIE's 11th Annu. Symp., Electronic Imaging '99: Security and Watermarking of Multimedia Contents*, vol. 3657, Jan. 1999.

[56] S. Thiemert, T. Vogel, J. Dittmann, and M. Steinebach, "A High-Capacity Block Based Video Watermark", *EUROMICRO 2004*, Page(s) 457-460.

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

[58] D. Simitopoulos, S. Tsaftaris, N. V. Boulgouris and M. G. Strintzis, "Compressed-domain Video Watermarking of MPEG Streams", *IEEE International Conference on Multimedia and Expo (ICME 2002)*, Volume 1, Page(s) 569 -572, Lausanne, Switzerland, August 2002.

[59] B. G. Mobasseri, "A Spatial Digital Video Watermarking that Survives MPEG", *IEEE International Conference on Information Technology: Coding and Computing*, March 27-29, 2000, Las Vegas.

[60] F. Petitcolas, R. Anderson, and M. Kuhn, "Attacks on copyright marking systems", *in Workshop on Information Hiding*, vol.1525 LNCS, pp.218-238, April 1998.

[61] D. Kirovski and F. A. P. Petitcolas, "Blind pattern matching attack on watermarking systems", *IEEE Transactions on Signal Processing*, vol.51, no.4, Page(s) 1045-1053, Apr 2003.

[62] M. Kutter and F. A. P. Petitcolas, "Fair evaluation methods for image watermarking systems", *Journal of Electronic Imaging*, Vol. 9, no. 4, Page(s) 445-455, October 2000.

[63] M. Kutter and F. A. Petitcolas, "A fair benchmark for image watermarking systems", *Proc. SPIE, Security and Watermarking of Multimedia Contents*, Vol. 3657, Page(s) 223-226, 1999.

[64] F. A. P. Petitcolas and R. J. Anderson, "Weaknesses of copyright marking systems", *Multimedia and Security Workshop at the 6th ACM International Multimedia Conference*, Pge(s) 55-61, England, 1998.

[65] I. J. Cox and J. P. M. G. Linnartz, "Some general methods for tampering with watermarks", *IEEE Journal on Selected Areas in Communications*, vol.16, no.4, Page(s) 587-593, May 1998.

[66] S. Voloshynovskiy and A. Herrigel, "Robustness of Watermarks and Intentional Attacks", *TCW Workshop on Digital Watermarking and Digital Rights Management*, Darmstadt, Germany, 25 and 26 September, 2002.

[67] S. Voloshynovskiy, S. Pereira, V. Iquise and T. Pun, "Attack modelling: towards a second generation watermarking benchmark", *Signal Processing, Special Section on Information Theoretic Aspects of Digital Watermarking*, Vol. 81, No 6, Page(s) 1177-1214, June 2001.

[68] S. Voloshynovskiy, A. Herrigel and T. Pun, "Blur/Deblur Attack against Document Protection Systems based on Digital Watermarking", *Proc. of the 4th International Information Hiding Workshop*, Pittsburgh, USA, April 2001.

[69] S. Voloshynovskiy, S. Pereira, T. Pun, J. J. Eggers, and J. K. Su, "Attacks on digital watermarks: classification, estimation based attacks, and benchmarks", *Communications Magazine, IEEE* , Vol.39, No.8, Page(s) 118-126, Aug 2001.

[70] S. Pereira, S. Voloshynovskiy, M. Madueo, S. M. Maillet and T. Pun, "Second generation benchmarking and application oriented evaluation", *Proc. of the 4th International Information Hiding Workshop*, Pittsburgh, PA, USA, April 2001.

[71] A. Herrigel, S. Voloshynovskiy, and Y. Rytsar, "The watermark template attack", *In 13th Annual Symposium, Electronic Imaging 2001: Security and Watermarking of Multimedia Content III*, SPIE Proceedings, San Jose, California USA, 23-27 January 2001.

[72] M. Kutter, S. Voloshynovskiy and A. Herrigel, "Watermark copy attack", *Proc of SPIE 12th Annual Symposium, Electronic Imaging 2000: Security and Watermarking of Multimedia Content II*, Vol. 3971, San Jose, California USA, Page(s) 23-28 January 2000.

[73] P. Comesana, L. Perez-Freire, and F. Perez-Gonzalez, "The return of the sensitivity attack", *in Proc. IWDW Conf*, Page(s) 260-274, Italy, September, 2005.

[74] M. E. Choubassi and P. Moulin, "A new sensitivity analysis attack", *in Proc. SPIE Conf.*, San Jose, CA, Page(s) 734-745, January 2005.

[75] F. Petitcolas and R. Anderson, "Evaluation of copyright marking systems", *Proc. IEEE Multimedia Systems'99*, Vol.1, Page(s) 574-579, June 1999.

[76] F. A. P. Petitcolas, M. Steinebach, F. Raynal, J. Dittmann, C. Fontaine, and N. Fats, "A public automated web-based evaluation service for watermarking schemes: StirMark Benchmark", *Proc of Electronic Imaging 2001, Security and Watermarking of Multimedia Contents*, Vol. 4314, San Jose, U.S.A., 22-26 January 2001.

[77] M. Kutter and F. A. P. Petitcolas, "Fair evaluation methods for image watermarking systems", *Journal of Electronic Imaging*, Vol. 9, no. 4, Page(s). 445-455, October 2000.

[78] F. Petitcolas, R. Anderson, and M. Kuhn, "Attacks on copyright marking systems", *Workshop on Information Hiding*, Vol.1525 LNCS, Page(s).218-238, April 1998.

[79] D. Kirovski and F. A. P. Petitcolas, "Blind pattern matching attack on watermarking systems", *Signal Processing, IEEE Transactions on*, Vol.51, No.4, Page(s). 1045-1053, Apr 2003.

[80] S. Voloshynovskiy and A. Herrigel, "Robustness of Watermarks and Intentional Attacks", *TCW Workshop on Digital Watermarking and Digital Rights Management*, Sept 2002.

[81] S. Voloshynovskiy, S. Pereira, V. Iquise, and T. Pun, "Attack modelling: towards a second generation watermarking benchmark", *Signal Processing, Special Section on Information Theoretic Aspects of Digital Watermarking, 2001*, Vol. 81, No 6, Page(s). 1177-1214, June 2001.

[82] S. Voloshynovskiy, A. Herrigel, and T. Pun, "Blur/Deblur Attack against Document Protection Systems based on Digital Watermarking", *Proc. of the 4th International Information Hiding Workshop*, Apr 2001.

[83] S. Pereira, S. Voloshynovskiy, M. Madueo, S.Marchand-Maillet, and T. Pun, "Second generation benchmarking and application oriented evaluation", *Proc. of the 4th International Information Hiding Workshop*, Apr 2001.

[84] S. Voloshynovskiy, S. Pereira, T. Pun, J. J. Eggers, and J. K. Su, "Attacks on digital watermarks: classification, estimation based attacks, and benchmarks", *Communications Magazine, IEEE* , Vol.39, No.8, Page(s).118-126, Aug 2001.

[85] F. A. P. Petitcolas, "Watermarking schemes evaluation", *Signal Processing Magazine, IEEE* , Vol.17, No.5, Page(s).58-64, Sep 2000.

[86] K. Jung, K. I. Kim, and A. K. Jain, "Text Information Extraction in Images and Video: A Survey", *Pattern Recognition*, Volume 37, Issue 5,Page(s) 977-997, May 2004.

[87] C. C. Lee, Y. C. Chiang, H. M. Huang, and C. L. Tsai, "A Fast Caption Localization and Detection for News Videos", *Second International Conference on Innovative Computing, Information and Control, 2007. ICICIC '07.*, Page(s) 226-226, 5-7 Sept. 2007.

[88] P. Shivakumara, Q. P. Trung, and L. T. Chew, "A Gradient Difference Based Technique for Video Text Detection", *Proceedings of 10th International Conference on Document Analysis and Recognition, ICDAR(2009)*, Page(s) 156-160, 26-29 July 2009.

[89] P. Shivakumara, T. Q. Phan, and T. C. Lim, "A Robust Wavelet Transform Based Technique for Video Text Detection", *Proceedings of 10th International Conference on Document Analysis and Recognition, 2009*, Page(s) 1285-1289, July 2009.

[90] C. Emmanouilidis, C. Batsalas, and N. Papamarkos, "Development and Evaluation of Text Localization Techniques Based on Structural Texture Features and Neural Classifiers", *Proceedings of 10th International Conference on Document Analysis and Recognition*, Page(s) 1270-1274, 26-29 July 2009.

[91] Y. Jun, H. Lin-Lin, and L. H. Xiao, "Neural Network Based Text Detection in Videos Using Local Binary Patterns", *Proceedings of Chinese Conference on Pattern Recognition, 2009*, Page(s) 1-5, 4-6 Nov. 2009.

[92] J. Zhong, W. Jian, and S. Yu-Ting, "Text detection in video frames using hybrid features", *Proceedings of International Conference on Machine Learning and Cybernetics*, Page(s) 318-322, 12-15 July 2009.

[93] S. Yu and W. Wenhong, "Text Localization and Detection for News Video", *Proceedings of Second International Conference on Information and Computing Science, 2009*, Page(s) 98-101, May 2009.

[94] C. W. Ngo and C. K. Chan, "Video text detection and segmentation for optical character recognition", *Multimedia Systems*, vol.10, No.3, Page(s) 261-272, Mar. 2005.

[95] M. Anthimopoulos, B. Gatos, and I. Pratikakis, "A Hybrid System for Text Detection in Video Frames", *Proceedings of The Eighth IAPR International Workshop on Document Analysis Systems*, Page(s) 286-292, 16-19 Sept. 2008.

[96] P. Shivakumara, T.Q. Phan, and T. C. Lim, "Video text detection based on filters and edge features", *Proceedings of IEEE International Conference on Multimedia and Expo, ICME(2009)*, Page(s) 514-517, June 28 2009-July 3 2009.

[97] P. Shivakumara, T.Q. Phan, and T. C. Lim, "An Efficient Edge Based Technique for Text Detection in Video Frames", *Proceedings of The Eighth IAPR International Workshop on Document Analysis Systems*, Page(s) 307-314, 16-19 Sept. 2008.

[98] P. Shivakumara, T.Q. Phan, and T. C. Lim, "Efficient video text detection using edge features", *Proceedings of 19th International Conference on Pattern Recognition, ICPR(2008)*, Page(s) 1-4, 8-11 Dec. 2008.

[99] Y. Su, Z. Ji, X. Song, and R. Hua, "Caption Text Location with Combined Features for News Videos", *Proceedings of International Workshop on Geoscience and Remote Sensing and Education Technology and Training*, Page(s) 714-718, 21-22 Dec. 2008.

[100] Y. Su, Z. Ji, X. Song, and R. Hua, "Caption text location with combined features using SVM", *Proceedings of 11th IEEE International Conference on Communication Technology.*, Page(s) 711-714, 10-12 Nov. 2008.

[101] U. Gargi, D. Crandall, S. Antani, T. Gandhi, R. Keener, and R. Kasturi, "A System for Automatic Text Detection in Video", *Proc. of International Conference on Document Analysis and Recognition.*, Page(s) 29-32, 1999.

[102] Y. K. Lim, S. H. Choi, and S. W. Lee, "Text extraction in MPEG compressed video for content-based indexing", *in Proc. Int. Conf. on Pattern Recognition*, vol. 4, Page(s) 409412, 2000.

[103] Y. Zhong, H. J. Zhang, and A. K. Jain, "Automatic caption localization in compressed video", *Proceedings of International Conference on Image Processing*, vol. 4, Page(s) 96-100, 1999.

[104] X. Qian and G. Liu, "Text Detection, Localization and Segmentation in Compressed Videos", *ICASSP, 2006*, Page(s) 385-388, 2006.

[105] Y. Zhong, H. Zhang, and A. K. Jain, "Automatic Caption Localization in Compressed Video", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.22, No.4, Page(s) 385-392 Apl. 2000.

[106] X. Qian and G. Liu, "Text Detection, Localization and Segmentation in Compressed Videos", *ICASSP, 2006*, Page(s) 385-388, 2006.

[107] X. Jiangbo, J. Xiuhua, and W. Yuxia, "Caption Text Extraction Using DCT Feature in MPEG Compressed Video", *WRI World Congress on Computer Science and Information Engineering, 2009*, vol.6, Page(s) 431-434, March 31 2009-April 2 2009.

[108] K. I. Trovato, "Method and apparatus for capturing broadcast EPG data for program title display", In *Patent Publication Number US 2004/6701526 B1*, filed on July 1, 1999.

[109] E. Esen, M. Soysal, T. K. Ates, A. Saracoglu, and A. A. Alatan, "A fast method for animated TV logo detection", *CBMI 2008*, Page(s) .236-241, June 2008.

[110] A. Ekin and E. Braspenning, "Spatial detection of TV channel logos as outliers from the content", *in Proc. VCIP*, SPIE, 2006.

[111] J. Wang, L. Duan, Z. Li, J. Liu, H. Lu, and J. S. Jin, "A robust method for TV logo tracking in video streams", *ICME, 2006*, ICME, 2006.

[112] N. Ozay and B. Sankur, "Automatic TV Logo Detection And Classification In Broadcast Videos", *EUSIPCO 2009*, Page(s) 839-843, Scotland, 2009.

[113] ISO/IEC 14496-10 and ITU-T Rec, H.264, Advanced Video Coding, *Draft of ISO/IEC and ITU-T*, 2003.

[114] T. Chattopadhyay, A. Sinha, and A. Hardikar, "H.264 Compressed Domain Watermarking In Content Delivery Network (CDN) Environment", *Proc. of the 2nd International conference on Computational Intelligence, Communication Systems and Network, (CICSyN2010)*, Page(s), 222-226, Liverpool, UK, 2010.

[115] T. Wiegand, G. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard", *Circuits and Systems for Video Technology, IEEE Transactions on*, Vol. 13, No. 7, Page(s) 560 - 576, July 2003.

[116] M. Horowitz, A. Joch, F. Kossentini, and A. Hallapuro, "H.264/AVC baseline profile decoder complexity analysis", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No. 7, Page(s) 704 - 716, July 2003.

[117] D. Chattopadhyay, A. Sinha, T. Chattopadhyay, and A Pal, "Adaptive Rate Control for H.264 Based Video Conferencing Over a Low Bandwidth Wired and Wireless Channel", *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, May 2009 .

[118] G. J. Sullivan and T. Wiegand, "Rate-Distortion Optimization for Video Compression", *IEEE Signal Processing Magazine*, Vol 15, Page(s) 74-90, Nov 1998.

[119] Joint Video Team (JVT)JVT-H014, "Draft of Adaptive Rate Control", *Draft of ISO/IEC MPEG & ITU-T VCEG*.

[120] S. M. Kay, "Fundamentals of Statistical Signal Processing: Detection Theory", *Prentice-Hall*, Inc. A Simon & Schuster, New Jersy 07458, 1998.

[121] D. Chattopadhyay, A. Sinha , and T. Chattopadhyay, "A low cost multiparty H.264 based video conference solution for corporate environment". *Proc. of the The International Conference on Computational Intelligence and Communication Networks (CICN2010)*, Nov 2010.

[122] T. Chttopadhyay and A. Pal, "Watermarking H.264", *http://www.embedded.com/columns/technicalinsights/202805174?_requestid=287915*, Embedded Design, Nov, 2007.

[123] "JM Reference H.264 Code", *http://iphome.hhi.de/suehring/tml/download/old_jm/*, Last Accessed on 25th January, 2011.

[124] S. R.Nirmala, S.Dandapat, and P. K. Bora, "Image quality assessment in retinal image compression systems", *IET-UK International Conference on Information and Communication Technology in Electrical Sciences (ICTES 2007)*, Page(s) 737-742, Dec. 2007.

[125] T. Poulus, "Connected TV allows operators to benefit from OTT content", *Telecom Paper*, Nov 02, 2009.

[126] A. Gonsalves, "Connected TV Sales Booming", *Information Week*, Aug 5, 2009.

[127] "Smart launches SmartBro Surf TV", *http://www.yugatech.com/blog/telecoms/smart-launches-surftv/*, Last accessed on Oct 2010.

[128] Press Release on, "SMART launches SurfTV", *http://smart.com.ph/corporate/newsroom/SurfTV.htm*, Last accessed on Oct 2010.

[129] A. Weiner, "CES Day 2: Yahoo!s Connected TV Looks Strong", *http://blogs.gartner.com/allen_weiner/2009/01/09/ces-day-2-yahoos-connected-tv-looks-strong/*, Last accessed on Oct, 2010.

[130] Research and analysiss for ddigital living technologies, "Fast Facts - Which connected TV features appeal most to consumers", *http://parksassociates.ecnext.com/coms2/summary_0256-11251_ITM*, Last accessed on Oct, 2010.

[131] H. McCracken, "The Connected TV: Web Video Comes to the Living Room", *http://www.pcworld.com/article/161565/the_connected_tv_web_video_comes_to_the_living_room.html*, Mar 23, 2009, Last accessed on Oct, 2010.

[132] Report on PC World, "Apple, roku and vodu", *http://www.pcworld.com/zoom?id=161565&page=1&zoomIdx=1*, Last accessed on Oct, 2010.

[133] Microsoft News Letter, "Microsoft Mediaroom Brings Connected TV to Life", *http://www.microsoft.com/presspass/press/2008/jan08/01-06MSMediaroomTVLifePR.mspx*, Last accessed on Oct, 2010.

[134] T. Chattopadhyay, A. Pal, and A. Sinha, "Recognition of Characters from Streaming Videos", *Sciyo Book*, Vol No Page(s) Sept. 2010.

[135] H. Ghosh, S. K. Kopparapu, T. Chattopadhyay, A. Khare, S. Wattamwar, A. Gorai, and M. Pandharipande, "Multimodal indexing of Multi-lingual News Video", *International Journal of Digital Multimedia Broadcasting*, vol. 2010, Article ID 486487, 18 pages, 2010. doi:10.1155/2010/486487.

[136] T. Chattopadhyay, A. Pal, and U. Garain, "Mash up of Breaking News and Contextual Web Information: A Novel Service for Connected Television", *Proc. Of ICCCN 2010 Workshop on Multimedia Computing and Communications (MCC 2010)*, Aug. 2010..

[137] T. Chattopadhyay and A. Sinha, "Recognition of Trademarks from Sports Videos for Channel Hyperlinking in consumer end", *Proc. of the 13th International Symposium on Consumer Electronics (ISCE'09)*, May. 2009.

[138] T. Chattopadhyay and A. Chaki, "Identification of Trademarks Painted on Ground and Billboards using Compressed Domain Features of H.264 from Sports Videos", *Proc. of the 2nd International conference on Computational Intelligence, Modeling and Simulation, (CIM-Sim2010)*, Sept. 2010.

[139] P. Vandewalle, S. Susstrunk, and M. Vetterli, "Lcav super-resolution source code and images, *http://lcavwww.epfl.ch/reproducible research/VandewalleSV05*, last accessed on Dec, 2010.

[140] wikipedia, "http://en.wikipedia.org/wiki/Electronic_program_guide", *wikipedia*, last accessed on Aug, 2010.

[141] Gorine and Andrei, "Programming Guide Manages Networked Digital TV", *http://www.eetimes.com/design/other/4012993/Programming-guide-manages-networked-Digital-TV?pageNumber=1*,December, 2002. Last Retrieved on Nov 8, 2010.

[142] T. Chattopadhyay, A. Sinha, A. Pal, D. Pradhan, and S. Roy Chowdhury, "Recognition of Channel Logos from Streamed Videos for Value Added Services in Connected TV", *Proc. of the 29th International Conference on Consumer Electronics (ICCE'11)*, Page(s) Jan, US, 2011.

[143] "http://en.wikipedia.org/wiki/Bhattacharyya_distance", *wikipedia*.