NOVEL SOURCE CODING METHODS FOR OPTIMISING REAL TIME VIDEO CODECS

A Thesis submitted for the degree of Doctor of Philosophy

By

Premkumar Elangovan

Department of Computing and Advanced Technologies,
Faculty of Enterprise and Innovation,

Buckinghamshire New University
Brunel University, West London

July, 2009

Abstract

The quality of the decoded video is affected by errors occurring in the various layers of the protocol stack. In this thesis, disjoint errors occurring in different layers of the protocol stack are investigated with the primary objective of demonstrating the flexibility of the source coding layer. In the first part of the thesis, the errors occurring in the editing layer, due to the coexistence of different video standards in the broadcast market, are addressed. The problems investigated are 'Field Reversal' and 'Mixed Pulldown'. Field Reversal is caused when the interlaced video fields are not shown in the same order as they were captured. This results in a shaky video display, as the fields are not displayed in chronological order. Additionally, Mixed Pulldown occurs when the video frame-rate is up-sampled and down-sampled, when digitised film material is being standardised to suit standard televisions. Novel image processing algorithms are proposed to solve these problems from the source coding layer. In the second part of the thesis, the errors occurring in the transmission layer due to data corruption are addressed. The usage of block level source error-resilient methods over bit level channel coding methods are investigated and improvements are suggested. The secondary objective of the thesis is to optimise the proposed algorithm's architecture for real-time implementation, since the problems are of a commercial nature. The Field Reversal and Mixed Pulldown algorithms were tested in real time at MTV (Music Television) and are made available commercially through 'Cerify', a Linux-based media testing box manufactured by Tektronix Plc. The channel error-resilient algorithms were tested in a laboratory environment using Matlab and performance improvements are obtained.

Acknowledgements

I would like to take this opportunity to thank my supervisors, Dr Peter Harding and Dr Gaoyong Luo, for their valuable supervision and support all these years. I am grateful to Professor John Boylan for supporting me with the funding necessary to carry out my research. I would like to express my gratitude to Professor Geoff Lawday for linking me with Tektronix and helping take over a commercial project as part of my research.

Especially, I would like to thank Tektronix for the support and funding they have provided and for accommodating me in their commercial laboratory for nine months. I would like to thank Dr. Paul Robinson, Dr. Mathew Bowers and Andy Hickman at Tekronix for the generosity shown towards my research project. I would also like to thank Oliver Barton, Sacha Darwin and Dr. Scott Griffiths at Tektronix for their valuable technical assistance throughout the project.

I would like to thank Professor Chris Hudson, Dr Peter Wilkinson, Laura Bray, Dr Anne Evans, Howard Bush, William Lishman, staff at Tektronix, staff at Bucks New University and my fellow research students for their help and support.

I dedicate this work to my beloved parents Elangovan Chinasamy, Meenalochani Elangovan and my sister Dr Janane Elangovan.

Author's Declaration

This is to certify that the work submitted in my thesis is my own and has not been submitted for any other degree.

Attention is drawn to the fact that some of the preliminary results have been published. Contents of Chapters 3, 4, 5 and 6 have been filed as patents:-

- **A Method of Quantifying Inter-field Motion in a Video Frame**, *Copyrights: Tektronix and Danaher Inc (2008).*

- **A Method of Identifying Inconsistent Field Dominance Metadata in a Sequence of Video**, *Copyrights: Tektronix and Danaher Inc (2008).*

- **A Method of Tracking a Region of Interest in a Sequence of Video Frames**, *Copyrights: Tektronix and Danaher Inc (2008).*

- **A Method of Determining Field Dominance in a Sequence of Video Frames**, *Copyrights: Tektronix and Danaher Inc (2008).*

- **Video Type Classification** *Copyrights: Tektronix and Danaher Inc (2008).*

Contents of Chapters 7, 8 and 9 have been published in conferences:-

- **Optimizing Video Codecs for Mobile Multimedia Applications**, *3rd IET European conference on Visual Media Production*, pp 182, London, (2006).

- **Structurally Efficient Video Codec for Wireless Mobile Applications**, *4th International Conference on Information Technology*, pp 190-195, Las Vegas, (2007).

- **Source Based Error Protection for Wireless Video by Motion Vector Sharing and Residual Garbaging**, *IEEE International Conference on Digital Signal Processing*, Cardiff, (2007).

- **Motion Vector Smoothing Algorithm for Robust Wireless Multimedia Communications**, *IEEE International Conference on Circuits and Systems for Communications,* Shangai (2008).

Contents

List of Illustrations

List of Tables

Abbreviations

3GPP	Third Generation Partnership Project
ACK	Acknowledgement
AIR	Adaptive Intra Refresh
AM	Acknowledgement Mode
AVC	Advanced Video Coding
AWGN	Additive White Gaussian Noise
BER	Bit Error Ratio
BLER	Block Error Ratio
BMA	Boundary Matching Algorithm
CABAC	Context Adaptive Binary Arithmetic Coding
CAVLC	Context Adaptive Variable Length Coding
CDMA	Code Division Multiple Access
CRT	Cathode Ray Tube
DCT	Discrete Cosine Transform
DVB	Digital Video Broadcasting
EDGE	Enhanced Data Rates for GSM Evolution
FDD	Frequency Division Duplex
GOP	Group of Frames
GSM	Global System for Mobile Communications
HDTV	High Definition Television
HEC	Header Extension Code
HVS	Human Vision System
IMT	International Mobile Telephony
JPEG	Joint Photographic Experts Group
MAD	Mean Absolute Deviation

KBPS	Kilo Bits Per Second
MBPS	Mega Bits Per Second
MBM	Motion Boundary Marker
MPEG	Motion Photographic Experts Group
NACK	Negative Acknowledgement
NCDMA	Narrow band Code Division Multiple Access
NTSC	National Television System Committee
OFDM	Orthogonal Frequency Division Multiplexing
PAL	Phase Alteration by Line
PBDBS	Partial Backward Decodable Bitstream
PER	Packet Error Ratio
PRVLC	Partial Reversible Variable Length Codes
ROI	Region of Interest
RVLC	Reversible Variable Length Codes
SAD	Sum of Absolute Deviation
SD	Standard Definition
SECAM	Sequential Colour with Memory
TDD	Time Division Duplex
UEP	Unequal Error Protection
UMTS	Universal Mobile Telecommunication Systems
VLC	Variable Length Coding
WCDMA	Wide band Code Division Multiple Access

Glossary

Frame	An image in video terms.
Field	A frame containing information only in alternate lines.
Resolution	Number of pixels used to represent a frame.
Spatial domain	Pixels that belong to the same frame.
Temporal domain	Pixels that belong to adjacent frames.
Refresh rate	Rate at which the frames are changed in the display.
Compression	Process of digitising video material using fewer bits.
Compression ratio	Size of the compressed video with reference to the original video.
Transcoding	Process of coding a video file from one compression to another.
Metadata	Special flags used for compressing the video.
Broadcaster	Responsible for production and delivery of commercial video to consumers.
Editing factory	Responsible for post production.
Manual eyeballing	Visually inspecting the video.
Interlaced	Video constituting fields.
Progressive	Video constituting frames.
Telecine	Process of digitising film material.
Pulldown	Process of increasing the frame rate.
Protocol stack	Stack of layers designed for a specific purpose.
False positives	Erroneous positive identification.
Interpolation	Reconstruction of missing data points.
Bursty errors	Errors occurring in clusters.
Redundancy	Unwanted or extra information.
Transmission	Propagating information across a medium.

Doppler frequency	Change in Frequency due to object motion.
Carrier frequency	Nominal frequency of a carrier wave.
Wavelength	Distance between wave repetitions.
Puncturing	Removal of some bits after coding.
Standard	Specification written for global usage.
Resynchronisation	Connecting back to the bitstream after an error.
Fading	Noise in mobile channels.

1. Introduction

1.1. Overview

In the era of digital multimedia, a video stream goes through a very long life cycle taking different formats, sizes and shapes. This journey, starting in a video camera, continues through editing houses where it is digitised for further processing. The digitised stream is then transcoded and wrapped into a suitable container for wired and wireless transmission. During the process, errors occur in the different layers, which have different levels of impact based on their location and intensity. Regardless of the nature and location of the errors, the final impact will be on the visual quality of the video stream. This thesis follows the journey of the video stream from its birth (capture) to death (display) and investigates some issues in the editing and transmission layers that affect the picture quality. Figure 1-1 shows the graphical representation of the video life cycle, starting from the capture stage to transmission and display.

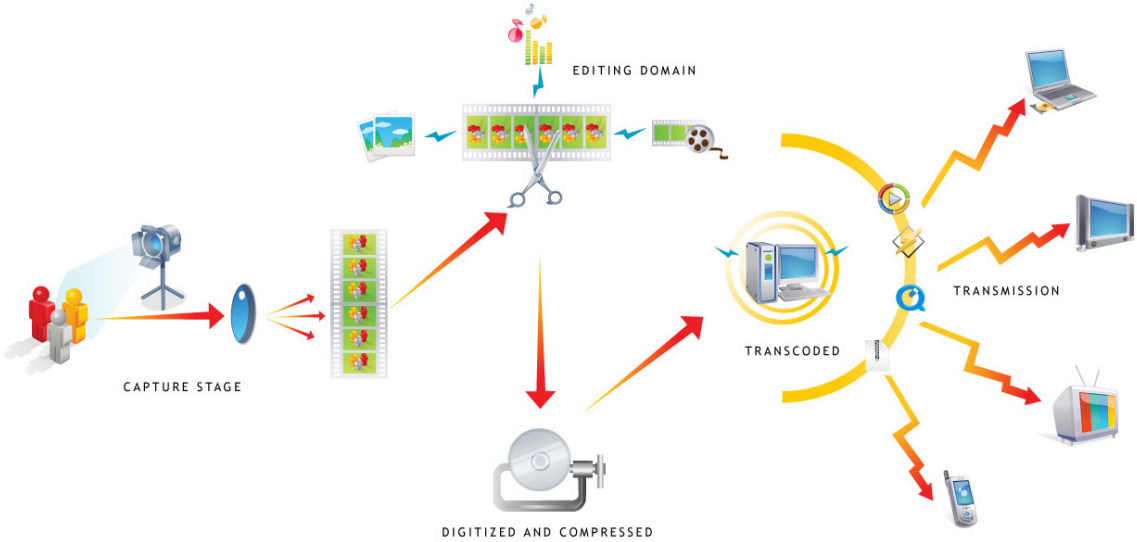


Figure 1-1. Video stream life cycle

1.2. Television and Displays

The video can be defined as a series of consecutive images displayed at a particular refresh rate or frame rate. Television was the first successful platform for displaying video. The era of television started when John Logie Baird transmitted video signals over the telephone line between London and Glasgow in 1927 (Genova and Levy, 2001). Subsequently, the television technology was standardised for analogue CRT (Cathode Ray Tube) displays. There are three standards in existence, namely, NTSC (National Television System Committee), an American standard; PAL (Phase Alternation by Line), a German standard; and SECAM (Sequential Colour with Memory), a French standard (Machida et al., 1979). When PAL was adopted as a European standard, SECAM lost its popularity, but it is still used in Russia and some East European countries. The influence of these three technologies is shown in Figure 1-2.

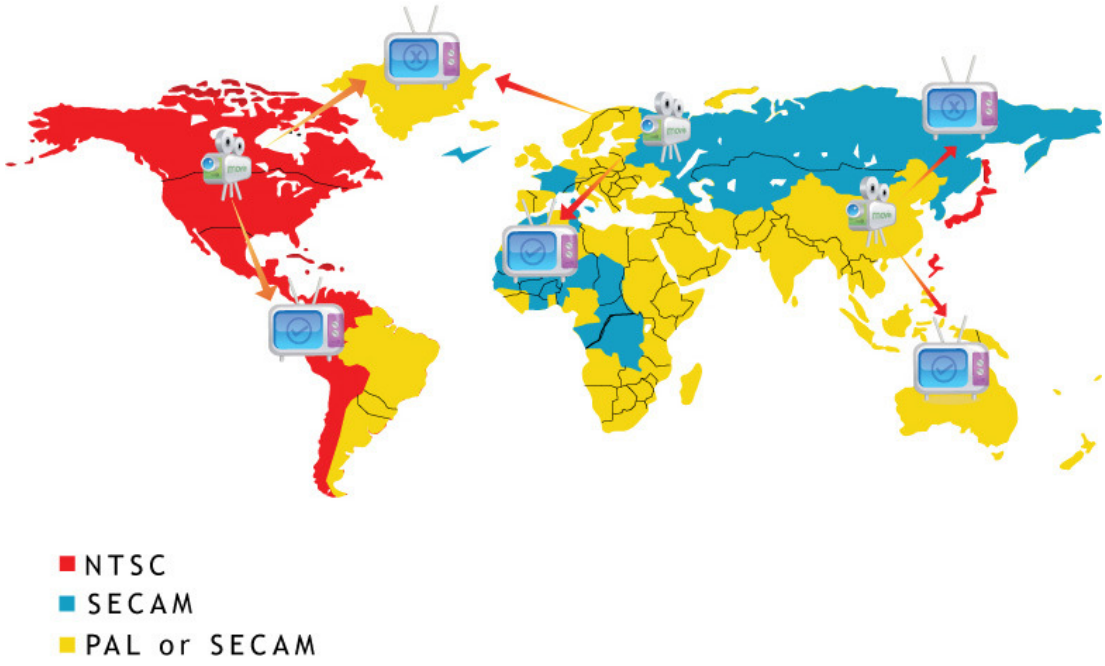


Figure 1-2. Compatibility issues with video captured in different formats (Abrams, 2009)

One of the main challenges researchers had to face while designing the circuitry of television in the early 20th century was to increase the scanning speed of the electron gun in the CRT. This led to the fundamental problem of maintaining the continuity of motion or persistence of vision. Originally, analogue electronics were capable of scanning approximately 200 lines every 1/50th of a second. This led to the introduction of interlaced video, in which each video frame contains either even or odd lines from the original image. These frames with reduced vertical resolution are termed as 'fields'. This reduced the amount of information transmitted for a frame by a factor of two and reduced the number of lines to be scanned for a frame, but in turn increased the complexity of the technology (Mallat, 2006). The loss in spatial resolution due to vertical sub-sampling is not visible due to the high refresh rate of the interlaced video (50-60 fields/sec).

Since then, the technology has evolved rapidly leading to the current transition from analogue to digital technology. The core technology behind analogue television is to deflect the electron beam emission from the electron gun present in the CRT. This illuminates the screen containing phosphor dots resulting in a video display by the fluorescent screen. The introduction of digital LCD (Liquid Crystal Display) displays led to the generation of flat screen displays. The LCD displays use the properties of liquid crystals in which the molecules of the crystals are aligned according to the voltage applied, resulting in the illumination of the corresponding pixel. The most recent advancement in flat screen display technology is 'Plasma', in which a mixture of gases trapped between glass panels is excited to illuminate the screen. O'Donovan (2006) provides a review of all existing display standards.

Advancement in electronics in the late 1960s meant that the scanning rate of the CRT circuitry could be increased. This advancement did not have a big impact on the interlaced standard, as any increase in the scanning lines per frame increases the bandwidth proportionally. When LCD monitors were introduced, the core technology was totally different from that of the CRT; the LCD paints the image on the monitor in a progressive fashion. The property of phosphorescence used in the interlaced screen was no longer applicable for the LCD displays. In the interlaced display, the odd lines are drawn first and the even lines are drawn next (it could also be even lines followed by odd lines). The even lines appear before the illumination of the odd

lines disappears to maintain the persistence in vision. In a progressive display, all the lines have to be drawn sequentially, which results in two fields captured in different time instants being displayed simultaneously. Jack (2004) presents comprehensive technical information on the interlaced and progressive standards in his book on television standards.

The progressive nature of the LCD displays led to the increase in progressive videos, which are designed exclusively for LCD screen technology. The coexistence of interlaced and progressive videos in the market has generated many quality and compatibility issues that require consideration. The two different protocols affect the quality of the displayed video in different ways, so it is difficult to pick the better of the two standards (Bruls and Ciuhu, 2005). This led to different interests in the broadcasting market. It was too late to question the interlacing technology due to its wide penetration in the market. The progressive displays captured the replacement market, and the large screens and magnified images attracted the consumers, in spite of the limitations.

The coexistence of interlaced video and progressive streams in the same video sequence has resulted in commercial broadcast videos that are ‘hybrid’ in nature. This coexistence led to a new field of research, in which a unique set of problems had to be dealt with. Figure 1-2 shows one of the compatibility issues with videos captured in different parts of the world; for example, a video captured in American NTSC format cannot be displayed in a European PAL television directly without undergoing a format conversion process.

Videos are captured at different frame rates; the frame rate may have to be up-converted or down-converted to suit diverse television standards. The popular addition to the hybrid video is up-converted video content using the ‘pulldown’ method. This process is applied to a video sequence generated by the ‘Telecine’ process, in which the film content is digitised (Childs, 1986). The pulldown process up-converts the frame rate of video content by inserting redundant fields, so that 24 frames/sec film video can be displayed on a commercial broadcast television functioning at a rate of 29.97 frames/sec. The speciality of the redundant frames generated during the pulldown process is that they resemble an interlaced frame in

appearance, as redundant frames contain a repeat field from previous or future frames. The pattern of the pulldown frames is expected to be uniform throughout the video sequence, which makes the inverse telecine process accurate, as the location of the redundant frames is known beforehand.

1.3. Video Compression

While progress was being made in television technology, two other technologies, 'video compression' and 'wireless' also underwent rapid developments. Image compression standards exploited the power of DCT (Discrete Cosine Transform) in the JPEG (Joint Photographic Experts Group) compression standard, which was later adapted to video compression (Pennebaker and Mitchell, 1993). The Motion Photographic Experts Group, which is abbreviated to MPEG, is an ISO/IEC standard. Though standardisation of digital video started with MPEG 1 (ISO/IEC 11172-2, 1993), which was preliminarily used for storage applications, MPEG 2 (ISO/IEC 13818-2, 2000) is considered to be the most successful standard, and had attracted the broadcasting market by 1995. The MPEG 4 (ISO/IEC 14496-2, 2001) standard exploits the object-based functionalities of video frames. MPEG 4 release 10, which is also known as H.264 (ISO/IEC 14496-10, 2008), is capable of producing better compression efficiencies compared to MPEG 4 at the expense of increased complexity.

The MPEG standards explained above were developed in parallel with H-Series standards, which targeted conferencing applications. The H.261 standard (ITU-T H.261, 1993) was intended for the video conferencing applications over ISDN lines. H.263 standard (ITU-T H.263, 2005) (which could be considered to be the superset of H.261), had added functionalities for modem-based communications. The latest addition to the digital video compression family is the MJPEG 2000 standard (ISO/IEC 15444-3, 2007), whose operating principle is based upon the wavelet transform. This standard could be considered as a new platform from the transform domain point of view and differs from DCT-based systems in technology. The multimedia frameworks, such as the MPEG 7 (Manjunath et al., 2002) and MPEG 21 (ISO/IEC 21000-1, 2004), are under development and are expected to be available by

2010. Figure 1-3 shows the various video compression standards in existence and their timeline.

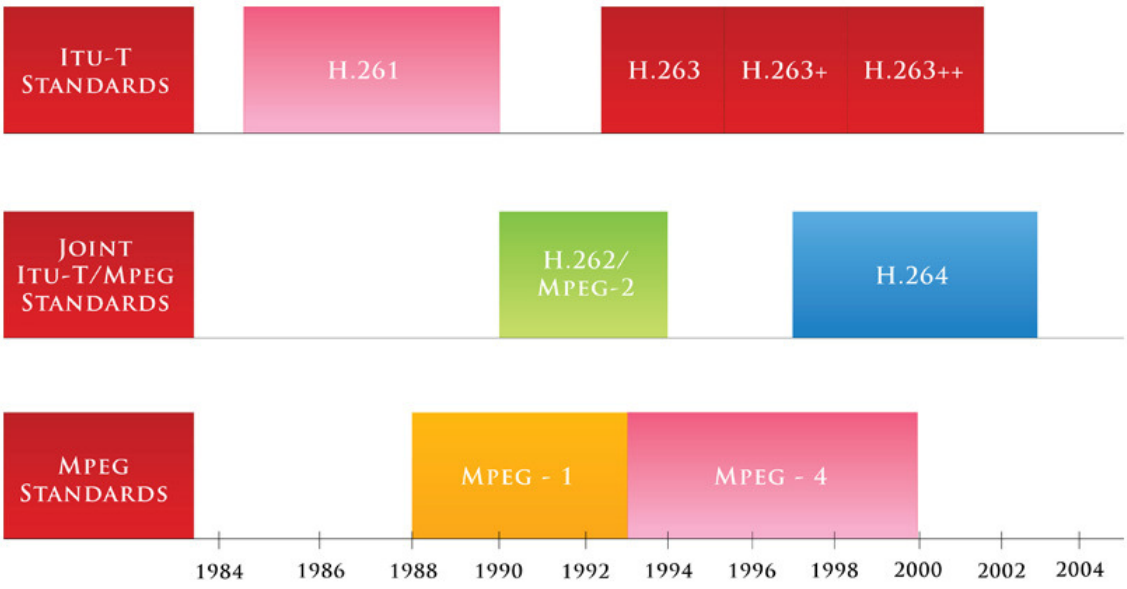


Figure 1-3. Video compression standards (Video Technology Magazine, 2009)

1.4. Wireless Standards

Though wireless systems had been in development for several years, the major innovation in wireless technology came in the form of GSM (Global System for Mobile Communications). The GSM standard is considered to be the second generation mobile communication system (Redl et al., 1995). The maximum voice data rate without coding for a user would be approximately 9.6 Kbps. The GPRS (General Packet Radio Service)/EDGE (Enhanced Data Rates for GSM Evolution) is an extension of the GSM system for providing data services (Halonen et al., 2003). The network architecture of the GPRS system is similar to that of the GSM system with some additional nodes. The GPRS is capable of providing data rates of 117 Kbps. The logic behind the high data rates is the multislot usage in the GSM system. The EDGE network is capable of providing higher data rates than GPRS, as EDGE utilises advanced modulation techniques and adapts them to the channel coding schemes. The maximum rate that EDGE can offer is about 483 Kbps.

The third generation wireless networks are based on the concept of CDMA (Code Division Multiple Access), which was originally standardised globally as IMT 2000 (International Mobile Telephony). The CDMA systems are also called UMTS (Universal Mobile Telecommunication Systems) in Europe, which operates on the principle of WCDMA (Wideband Code Division Multiple Access) with an operating bandwidth of 5MHz (Holma and Toskala, 2002). The CDMA systems are also called IS-95 in America, which operates on the principle of NCDMA (Narrowband Code Division Multiple Access) with an operating bandwidth of 1.25MHz. The IS-95 differs from the WCDMA in some aspects, but they are similar in principle and operation. The WCDMA transmission can reach data rates of up to 2 Mbps. The CDMA principle is applied to the system by means of a spreading sequence. The spreading sequence is a pseudorandom sequence which has good autocorrelation and cross correlation properties. The input sequence or data bits are multiplied by these codes to spread the signal across a 5MHz bandwidth. On the receiver side the data bits are retrieved by multiplying the sequence again with the same spreading code used at the transmitter.

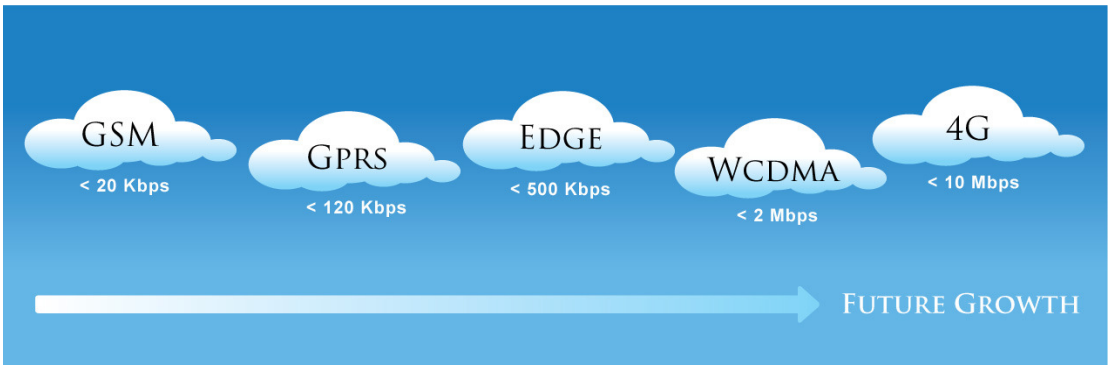


Figure 1-4. Mobile wireless standards

The latest addition to the wireless technology is DVB (Digital Video Broadcasting). This technology uses OFDM (Orthogonal Frequency Division Multiplexing) and MPEG-2 compression standard (Reimers, 2006). DVB-T is standardised for power-operated devices and DVB-H is standardised for battery-operated devices. The DVB-H uses advanced time slicing principles to reduce the

battery usage. This technology could be considered as the future of wireless multimedia technology (Rezaei et al., 2008).

1.5. Video Coding Issues

When video compression, wireless, and television standards were integrated together, there were many issues that affected the quality of the displayed video. This resulted in data being transmitted through multiple protocol stacks designed for different applications. This thesis approaches the video coding technology from a very broad perspective by investigating some blocks of the whole end-to-end architecture. The end-to-end architecture could be broken down into blocks; each block has a significant functionality and contributes in a unique way. For example, 'capture block' facilitates the capture of the raw video, 'editing block' edits different videos together, 'source coding block' is responsible for the video compression, 'channel coding block' adds redundancy for error detection, 'transmission block' transmits the video across wired and wireless channels, 'channel decoding block' corrects the errors that occur during transmission, 'source decoding block' decodes compressed video back into raw image frames, and finally, 'display layer' displays the image frames in progressive and interlaced screens.

In the video coding context, the end-to-end architecture is known as 'protocol stack' and blocks are known as 'layers'. The core theme investigated revolves around the following three issues: -

- The final impact will be on the display quality of the video, regardless of the location of the errors in the layers of the protocol stack.
- Some layers in the protocol stack do not function effectively, as they are not specifically designed for 'video' application.
- Only modifications incorporated in certain layers of the protocol stack can be implemented in real-time, because most of the layers are standardised for global usage.

The efficiency of the video coding algorithms can be assessed by following two factors: -

- The quality of the video measured using either subjective or objective metrics.
- Ease of implementation in real time and integration into existing architecture without modifying a standardised protocol stack.

The ‘source coding layer’ is the most flexible for real-time implementation of all the layers and also the video data are highly meaningful in this layer. Two disjoint problems that occur in different layers of the protocol stack were chosen for this specific research and the investigation was carried out to find an improved solution from the ‘source coding layer’.

Chapters 3-6 investigate problems that occur in the ‘editing layer’. The problems are currently addressed in the ‘display layer’. The ‘field reversal’ and ‘mixed pulldown’ problems occur in the editing layer of the protocol stack due to unskilled editing and the coexistence of different video standards. The current solution to the problem is to inspect the erroneous clips visually by playing them on a screen; this happens in the ‘display layer’. The problems were reported to Tektronix, which manufactures equipment for compressed video quality analysis by broadcasters like Google, MTV, Disney and Microsoft. The ‘field reversal’ occurs when the fields are not shown in the same order as they were captured. This is caused by editing together video materials that were captured in different television formats and it results in a ‘shaky video display’. The mixed pulldown problem is caused when the pattern of insertion of the redundant frames during the pulldown process (frame rate up-sampling by redundant frames) does not follow a uniform pattern. This is primarily caused by improper editing of two video sequences with different pulldown patterns. This results in serious errors in the reverse telecine process, as the reverse telecine assumes the pattern to be uniform and removes original frames instead of redundant frames.

The video stream loses its integrity and robustness, leading to a decrease in the display quality and the compression ratio, when hybrid sequences fail to use the metadata in the compressed bitstream to convey interlaced, progressive and pulldown

information. Figure 1-5 shows the graphical representation of different video types coexisting in the same stream. In the first section of the thesis, novel source coding methods to rectify the editing layer errors, which serve as an alternative to the exhaustive visual inspection process in the display layer, are proposed, to improve the overall picture quality.

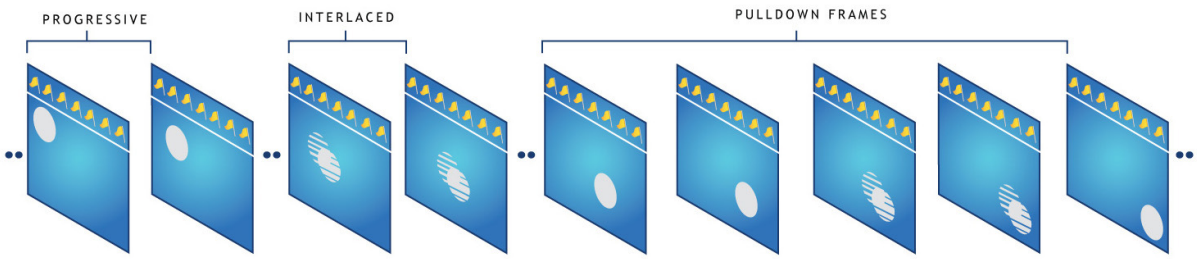


Figure 1-5. Different Video Types

Chapters 7-9 investigate problems that occur in the 'transmission layer' due to data loss in the transmission channel as a result of signal fluctuations. In spite of dramatic bandwidth increase in wireless channels, the channels tend to remain noisy and bursty (errors occurring in clusters), and their characteristics follow probability distributions. The unpredictable nature of the mobile channels results in severe fading effects, which can destroy the integrity of streaming multimedia applications. The problem is currently addressed by adding extra redundancy in the 'channel coding layer' and then utilising the redundant information to rectify errors in the 'channel decoding layer'. Figure 1-6 illustrates the data loss occurring in the channel.

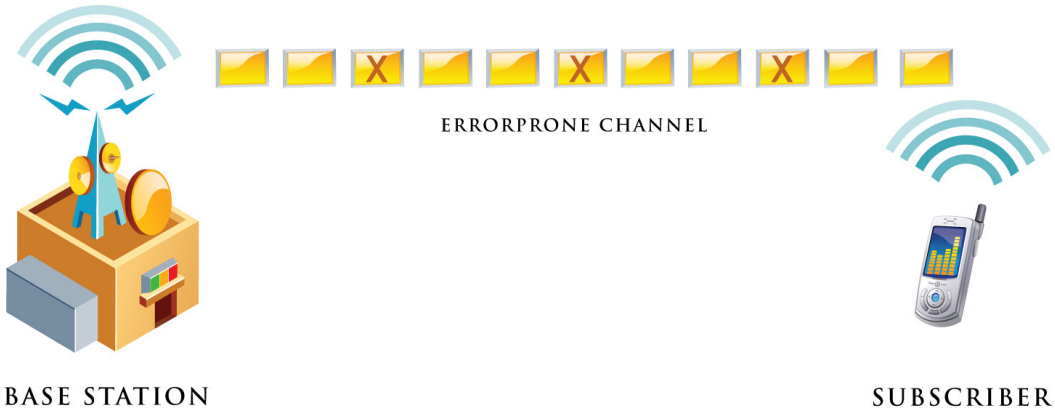


Figure 1-6. Data loss in the channel

In the second section of the thesis, subsequent to the investigation, the suitability of the source coding methods over channel coding methods for rectifying data losses are justified. This is followed by a demonstration of some novel source error control methods for robust mobile multimedia transmission.

1.6. Aim and Hypothesis of the Research

The aim of the research is to address errors occurring in the different layers of the video transmission system and propose novel methods to resolve the issues in order to improve the picture quality that are suitable for implementation in a real-time setting. Though this thesis addresses a wide variety of issues that are disjoint, the aim of the thesis is to test the following hypothesis: -

Regardless of the location and the nature of the occurrence of the errors in the compression video transmission system, a more efficient solution for the problems may be obtained from the source coding layer, as it may offer more flexibility than any other layer in the protocol stack.

Figure 1-7 shows the graphical representation of the above hypothesis. The errors occurring in the editing and transmission layers are addressed and effective solutions are proposed from the source coding layer.

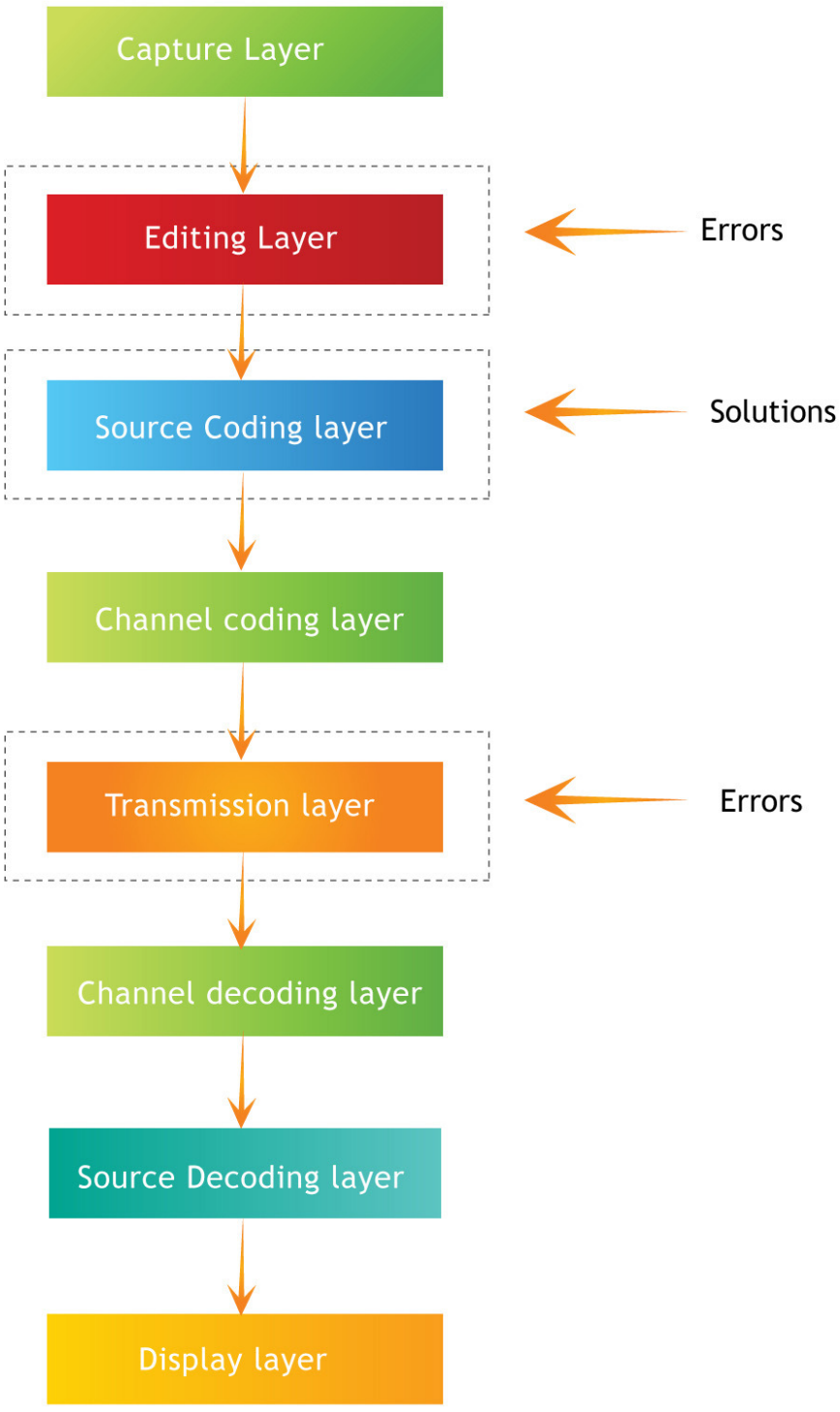


Figure 1-7. Thesis hypothesis

1.7. Organisation of the Thesis

The thesis has ten chapters and they are organised as follows: -

Chapter 2 presents technical background information on television, video compression and wireless standards. The next chapter investigates problems such as field reversal, mixed pulldown, and channel data loss occurring in editing and transmission layers. The chapter concludes with explanation on the research techniques used in the investigation.

Chapter 4 explains the need for a robust inter-field quantifier for interlaced-progressive frame classification. The drawbacks of the existing system are explained by using frequency domain analysis. The design process of a novel inter-field quantifier is explained step-by-step by incorporating object segmentation and representation methods. The next chapter presents the design process of novel field reversal and mixed pulldown detection algorithms. The inter-field quantifier from Chapter 4 is integrated with the designed algorithms and the performance improvement is justified. An optimisation procedure, the aim of which is to reduce the complexity of the final algorithm to suit real time hardware, is explained. In the following chapter, performance of the methods proposed to rectify editing layer errors in Chapters 4 and 5 are presented using simulation results.

Chapter 7 emphasises the importance of source error control methods over channel error control methods. This is backed up with novel data hiding methods. In the next chapter, the discussion from the previous chapter is extended to test the hypothesis on a well known source error resilient method, namely, the 'resync marker'. A novel method of 'virtual partitioning' is explained, in which the bitstream is synchronised without using any bits physically in the bitstream, despite emulating the functionality of the resync marker. In the following chapter, the performance of the methods proposed for rectifying transmission layer errors in Chapters 7 and 8 are presented using simulation results.

Chapter 10 concludes the thesis. The review of the work presented in the thesis is summarised, highlighting the contribution to the body of knowledge. The results

from all the chapters are summarised to establish the primary hypothesis of the thesis. This chapter also suggests a possible future direction of the research.

1.8. Publications and Patents

The general content of all the subsequent chapters in the thesis has been published in some form. The first section of the thesis (Chapters 3-6) represents the commercial work done in collaboration with Tektronix and is protected with five patent filings. The patents are given in Appendices C, D, E, F and G. The second part of the thesis (Chapters 7-9), which is more theoretical in nature, has been published at four international conferences. The published works are referenced at appropriate sections in the thesis.

1.9. Summary

This chapter summarised the entire video broadcasting system and gave the problem definition and objectives of the research. The growth of all the core technologies including television, compression and wireless protocols were reviewed. When core technologies with different protocol stacks designed for specific applications are challenged to handle applications for which they are not specifically designed, then this results in errors. These errors may be caused by either unskilled editing or transmission noise. In this thesis, methods for improving the video quality are proposed, and subsequently, the hypothesis to establish that of all the layers, the source coding layer is the best layer to resolve the errors occurring at any layer of the protocol stack is proven.

2. Literature Review

2.1. Introduction

This chapter gives a background introduction to the video protocol stack. The discussion starts by reviewing television standards and continues through video compression principles and fading channel characteristics. Section 2.2 explains the popular television standards along with general brief information on digital television technology and screens. The characteristics of interlaced, progressive and pulldown videos are presented in section 2.3. The following section explains the technology behind video compression. Section 2.5 explains the fading characteristics of the wireless mobile channels, and subsequently, a discussion on the suitability of various channel models for the research is presented. Since the research domain is very wide, the literature review is restricted to topics in various layers that are relevant to the research.

2.2. Television Standards

Standard resolution PAL and NTSC systems have an active visible video resolution of 720 x 576 and 720 x 480 respectively. The PAL standard has a high vertical resolution and a low refresh rate (25 frames/sec or 50 fields/sec); the NTSC has a low vertical resolution and a high refresh rate (29.97 frames/sec or 60 fields/sec). The resolution of the initial television standards were carefully selected based on the bandwidth and quality constraints (Ashbridge, 1937). For a standard television, each scan line lasts for approximately 64μs, which allows 858 pixels to be sampled at a rate of 13.5 MHz. Deducing the time for horizontal and vertical blanking will result in an active horizontal resolution of 720 pixels.

Video was initially represented in the 'RGB' format; due to bandwidth limitations and the need for a composite signal that suited both colour and monochrome terminals, colour difference signals 'YCbCr', 'YUV' and 'YIQ' were introduced. Human eyes are less sensitive to colour signals, so a low sampling rate

was used to transmit the colour signals (Wong and Bishop, 2006). The NTSC and PAL standards use YIQ and YUV respectively, with the colour signals sampled at a rate of 6.75MHz. This resulted in the generation of chroma planes (planes with colour information) with each line extending 320 pixels. Figures 2-1 and 2-2 show the graphical representation of PAL and NTSC television standards respectively.

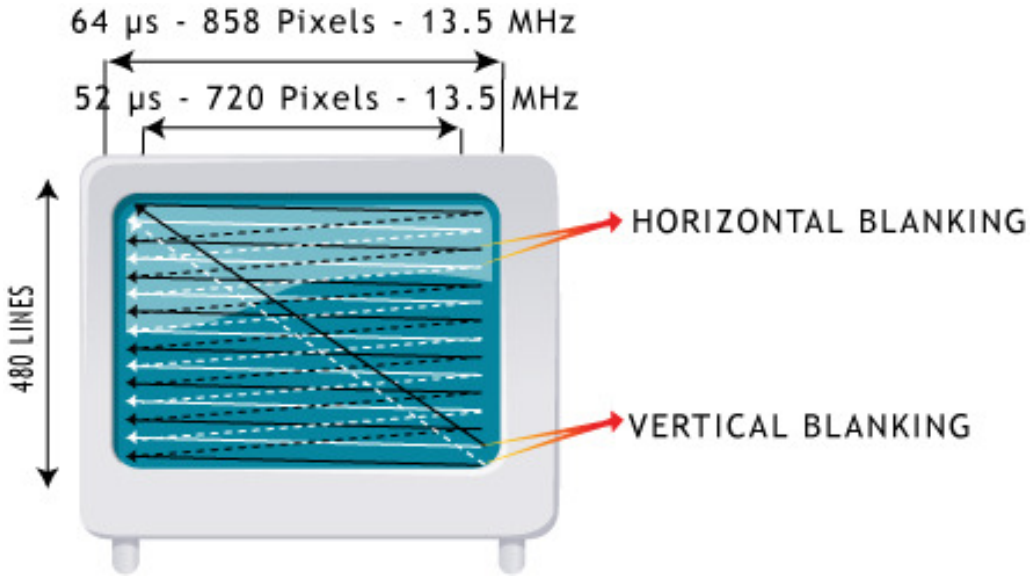


Figure 2-1. PAL standard television

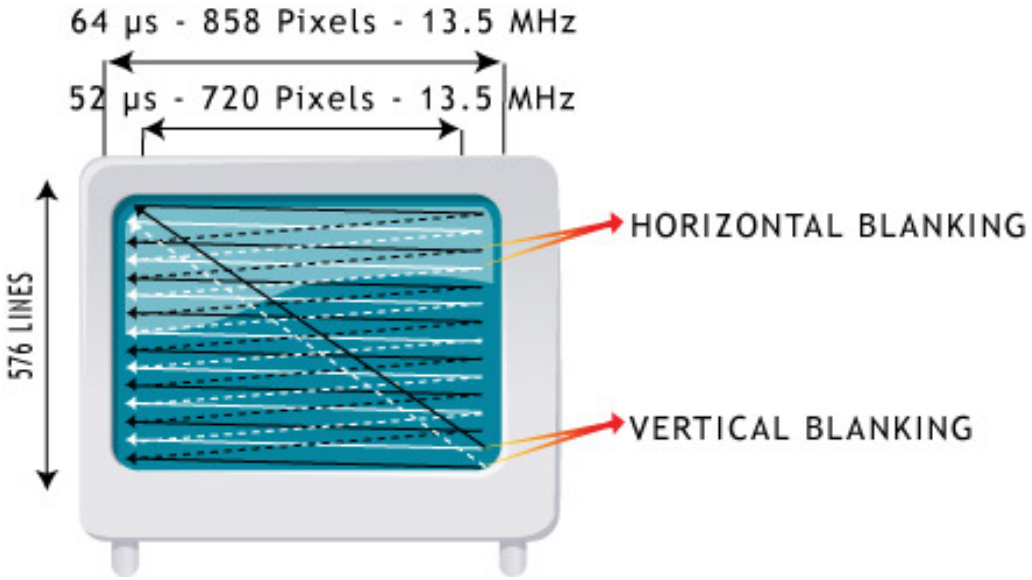


Figure 2-2. NTSC standard television

The 'aspect ratio' of a video is the ratio of image width to image height. The most popular aspect ratio used in standard televisions is 4:3. The technology

advancement in displays led to an increase in screen size, as did the aspect ratio. The aspect ratio 16:9 is more common nowadays with flat screen televisions. Other aspect ratios are used for the convenience of the applications (1.85 and 2.39:1 for movies, 5:4, 7:5 and 1:1 for still photography). The diverse usage of aspect ratios in the video source and the rapid increase in the display size led to the fundamental problem of compatibility; videos viewed in different screens were no longer the same. The video had to be letterboxed or pillar-boxed (shrinking the image and filling empty space by matting to avoid bad stretching) before being displayed (Deng et al., 2008). Expensive displays used efficient up-scaling methods to fit a low resolution video onto a high resolution display without diluting the spatial detail of the stream.

Increasing the image size by up-scaling dilutes the image details, and the clarity of the image is lost. To solve the problem, HDTV (High Definition Television) has been introduced by various television channels, where the bit rate of the digital transmission is increased by increasing the sampling rate for better picture quality (Navarro, 2002). The challenge is to handle the increase in bandwidth, as the screen resolution is almost tripled. One HDTV channel will take the space of three SD (Standard Definition) channels. The accepted standards for HDTV are 1280 x 720 (720p60) progressive at 60 fields per sec and 1920 x 1080 (1080i50) interlaced at 50 fields per sec. This migration leads to new generation television and results in sharper and better quality picture, despite being backward compatible with existing equipments (televisions with no HD add-on will play the basic SD layer of the transmission).

2.3. Interlaced and Progressive Videos

The interlaced and progressive videos differ in spatial and temporal characteristics and the choice of either format for a specific application is controversial (Vandendorpe and Cuvelier, 1999). A progressive video displayed on an interlaced display will have a low spatial resolution. In spite of the low temporal resolution experienced while displaying a progressive video on an interlaced display, the refresh rate is close to 30 frames/sec. Above this refresh rate, it is very unlikely that human eyes would be able to perceive the temporal variations. Displaying a

progressive video on an interlaced display produces acceptable results in both temporal and spatial domains, whereas displaying interlaced video on a progressive display became a challenging research problem because of the variations in the temporal and spatial characteristics.

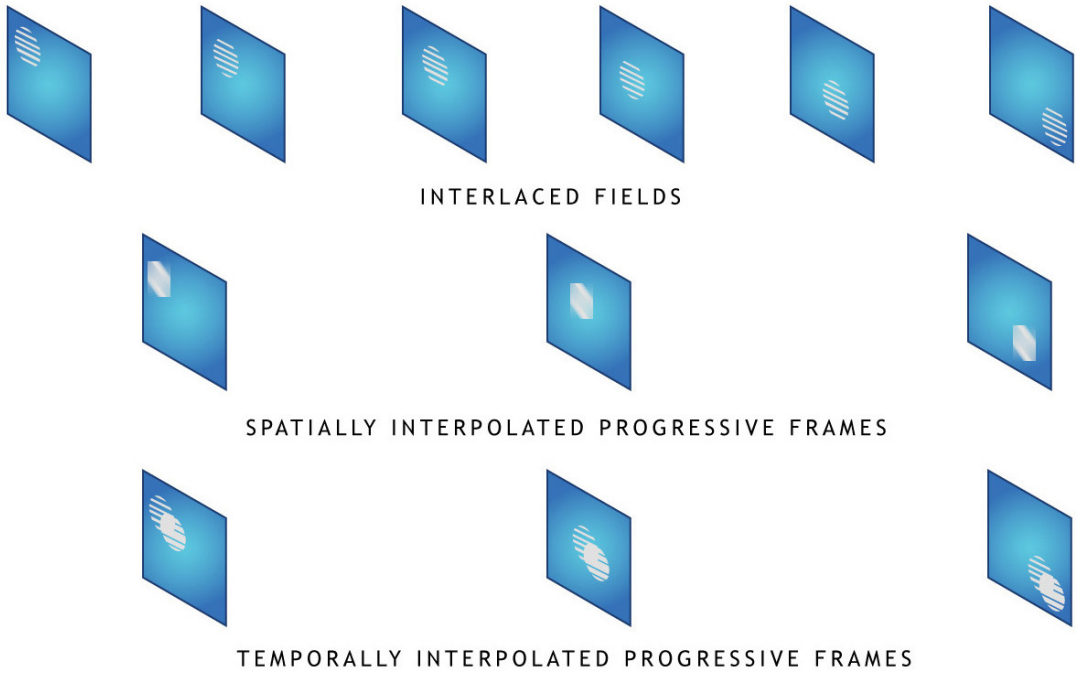


Figure 2-3. De-interlacing methods

When an interlaced video has to be displayed on a progressive screen, some operations have to be undertaken to up-sample (interpolating to high resolution) an interlaced field into a progressive video frame. This process is called de-interlacing (De Haan and Bellers, 1998). There are many de-interlacing methods in existence adopted by various display manufacturers. This process is mandatory because of the difference in display protocols between progressive and CRT screens. In short, the missing lines of the interlaced field have to be filled up in such a way that the smoothness in the picture is maintained. The big challenge in performing this operation is that the successive fields to be up-sampled belong to different time instants. The spatial interpolation will give good results if there is motion between the fields, whereas temporal interpolation will give good results if there is no motion happening between the fields. In practice, perfect de-interlacing cannot be achieved

as it is a non-linear problem (Mallat, 2006). This increases the burden on the quality of the de-interlacing algorithms, but the trade off between quality and the complexity demands advanced hardware for implementation. Different de-interlacing methods are shown in Figure 2-3.

'Pulldown' is a process of speeding a film-based telecined 24 frames/sec video content to 29.97 frames/sec NTSC or 25 frames/sec PAL by adding redundant frames in a uniform pattern all through the stream. For NTSC conversion, the video is slowed down to 23.976 (24000/1001) frames/sec and four frames of the source video are converted to five frames of the interlaced video. This is achieved by converting the four frames to the equivalent eight fields (top and bottom pairs) and then generating ten fields by duplication. The 3:2 pulldown refers to the method of duplicating fields once or twice every five frames (NTSC) and 12:2 pulldown refers to duplicating fields once or twice every 12 frames (PAL).

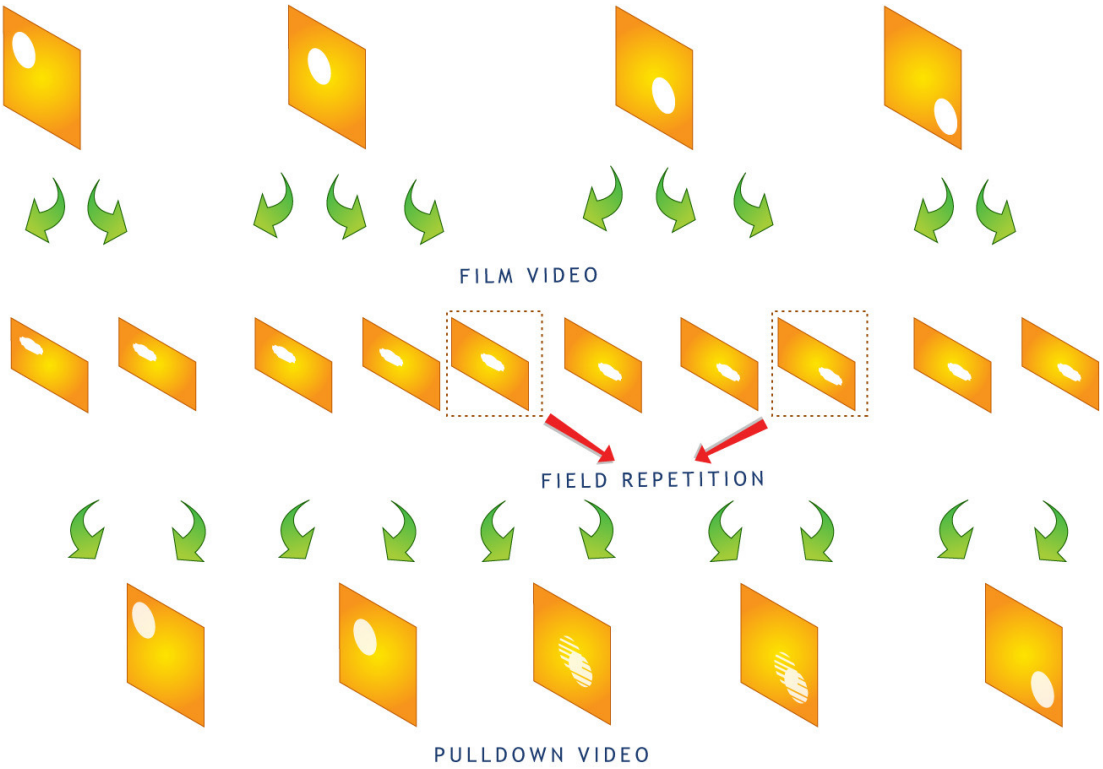


Figure 2-4. Pulldown process

The pulldown process results in new frames having combing artefacts (artefacts observed in interlaced frames due to the time difference between the fields), since the pairs of fields in these frames come from different frames rather than from the same frame (Nicolas et al., 2008). The combing artefacts are shown in Figure 2-5 and the pulldown process is illustrated in Figure 2-4. The reverse telecine is the opposite process, where the redundant frames are removed and the original 24 frames/sec material is reconstructed to make it suitable for progressive display (Wells, 1998).

Progressive frame **Interlaced/Pulldown frame**

Figure 2-5. Combing artefacts

2.4. Video Compression

Video compression is the process of reducing the number of bits required to represent the information in a digitised format. Video compression contains a very complicated hierarchy of layers (Tudor, 1995; Haskell et al., 1996). The video sequence could be considered as a running block of images that is refreshed at a particular display rate. The recommended rate is 25 frames per second, which avoids flicker. Each image is known as a frame and each frame is partitioned into a number of slices. In turn, the slices are partitioned into a number of macro-blocks (MB). Each macro block contains a single luminescence block of size 16 x 16 pixels and two chrominance blocks of 8 x 8 each; the former contains the brightness information and the latter contains the colour information. The chrominance blocks are sub-sampled to a lower resolution because the human eye is not very sensitive to colour signals. It is not necessary that the macro-blocks should follow the same 16 x 16 and 8 x 8 pixel matrix sizes; the sizes vary with the standard. For example, in

H.264, the macro-block takes sizes of 4 x 8, 4 x 4, 8 x 16 and so on. The video coding hierarchy is shown in Figure 2-6.

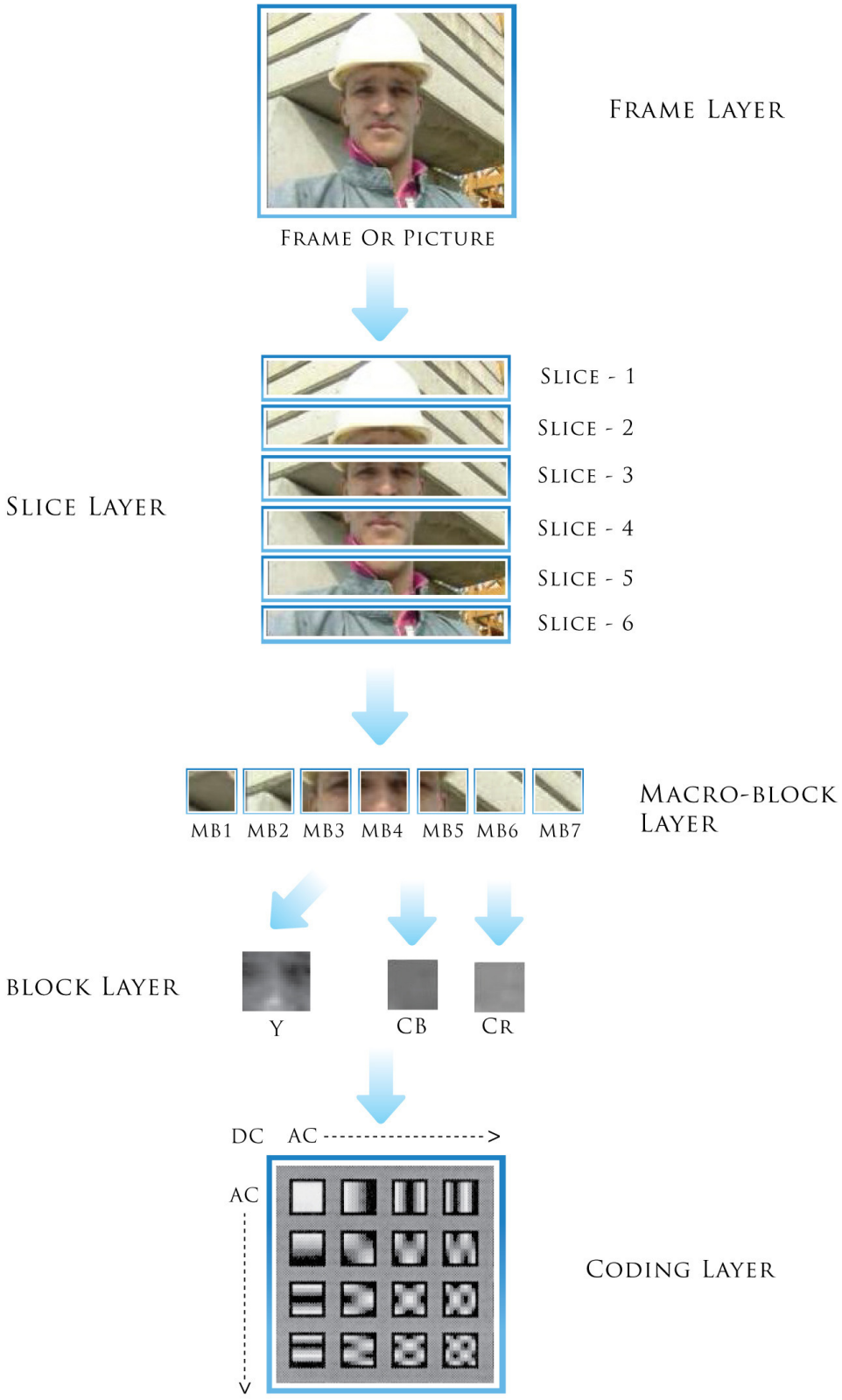


Figure 2-6. Video coding hierarchy

There are three kinds of frames used in compressing the video sequence: intra predicted (I), inter-predicted (P) and bidirectional-predicted (B) (Hanzo et al., 2007). The I frame, which is also known as an information frame, is always the first frame of any video compression system. The coding methodology of the I-frame is similar to that of the JPEG algorithm. The frame is partitioned into multiple blocks and each block is transformed by the DCT into frequency components (Balkanski et al., 1994). Transforming the time coefficients into frequency coefficients can be accomplished using many mathematical transforms, but the unique nature of the DCT transform is that the resulting frequency coefficients are floating point real values, whereas other transforms, like the Fourier transform, represent frequency coefficients as complex values. The 'I' frames are key frames, as they are not predictive coded (compressing the difference between successive values); their bit consumption is much higher than other frame types.

The P-frames use a different methodology for compressing the frames. The P frames are predictive coded frames and they rely upon the previous frame for reconstruction. The basic techniques utilised are motion estimation and compensation. In motion estimation, each macro-block of the current frame is compared with previous successfully reconstructed frame and searched for a close match. The address or location of the best matching area is coded as a motion vector. There are many block matching algorithms that can be utilised to reduce the number of iterations in finding the best match (Su and Sun, 2006; Lu and Tourapis, 2005).

In order to produce a 'P' frame, the best matching block from the previous frame is subtracted from the current block and the residual values are coded. In the receiver side, the decoder uses the motion vectors to identify the best matching block in the reference frame, and the residual values are added to reconstruct the block of the current frame, this process is motion compensation. The B frame's coding structure is identical to that of the P frame with the exception that it may use both future and previous frames for motion vector estimation; in other words, the blocks could use either the future frame or the current frame as the reference. The B frames uses minimal bits and maximum complexity in comparison with I and P frames. Since the B frames need both current and future frames for reconstruction, the coding order is different from the display order, which is shown in Figure 2-7.

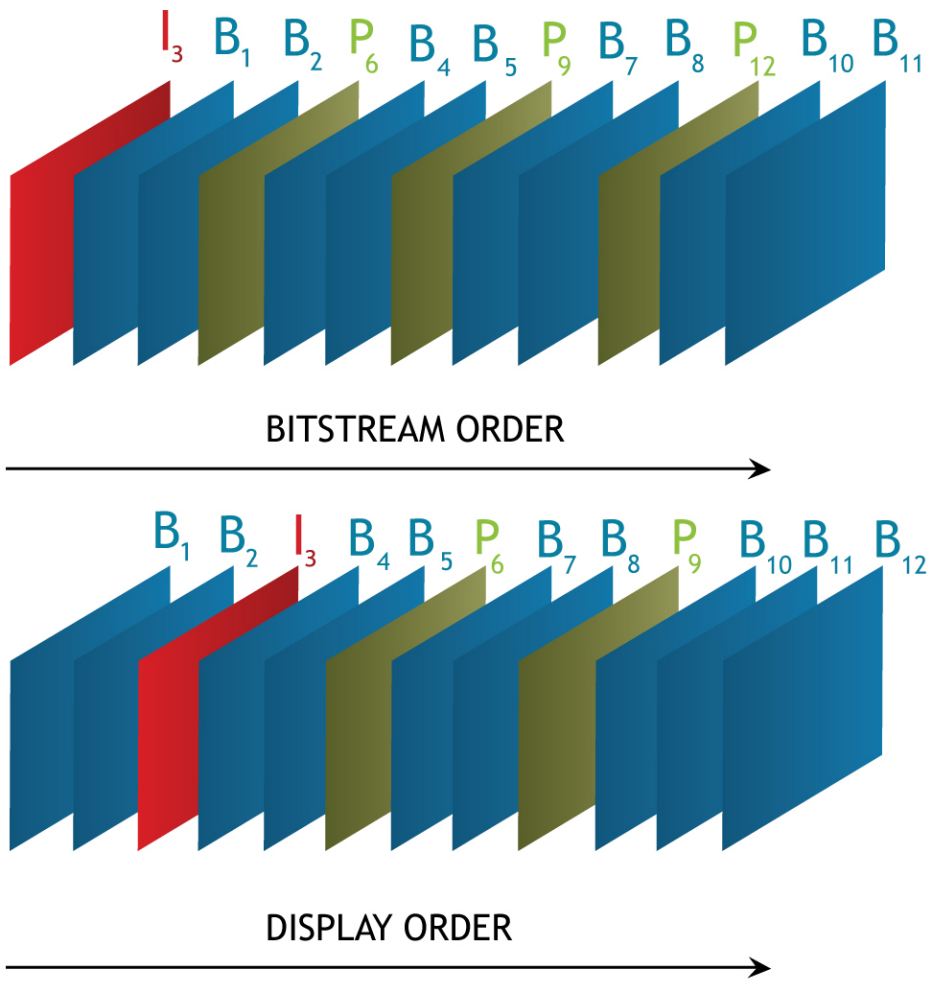


Figure 2-7. Coding and display order

The frequency coefficients are quantised by a matrix and a scaling factor. The low frequency components cluster on the top left corner and the values gradually decrease in frequency moving towards the bottom right. Quantisation is the process of reducing the high frequency information and preserving the dominant low frequency components (Richardson, 2004). This process of quantisation has a direct impact on the compression ratio; if the quantisation is 'coarse', then there will be an increase in the compression ratio, but a reduction in the quality of the image. If the quantisation is 'fine', then there will be a decrease in the compression ratio, but the quality of the compression image will be perceptually equal to the original. Quantising an image with low variance or constant texture will result in less

objective distortion than quantising an image with high variance or high spatial detail (Yang et al., 2005). The first frequency coefficient is known as the DC component and other coefficients are known as AC components.

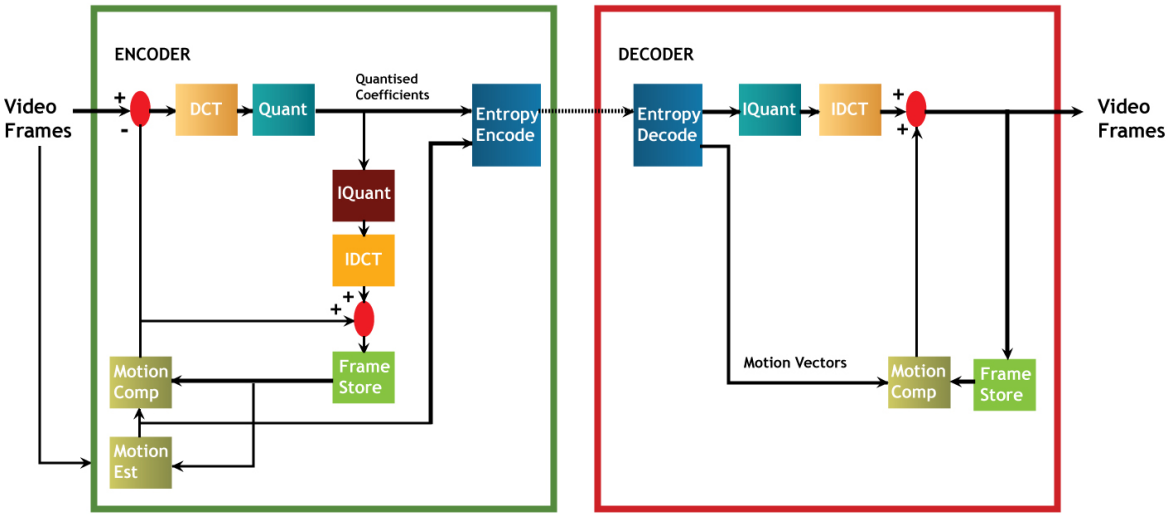


Figure 2-8. Video compression system (Richardson, 1999)

The DC component is compressed using differential coding, as it represents the average brightness of the block, whereas the AC coefficients are coded using run length coding and they represent the spatial variations of the block. The same process is applied to B and P frames with some variations. The quantisation matrix for P and B frames differs from that of I frame and there is no separate coding method for representing DC and AC coefficients, so all the coefficients are run-length coded. The coded coefficients are represented in the compressed bitstream using VLCs (Variable Length Coding), where the binary codes are of different lengths based on their entropy (average information based on the frequency of occurrence) (Sullivan and Wiegnad, 2005). The graphical representation of the video compression system is shown in Figure 2-8.

This thesis will focus on MPEG-based standards, as they are more flexible and more effective and support diverse applications. The basic structure remains the same for MPEG 2 (ISO/IEC 13818-2, 2000), MPEG 4 (ISO/IEC 14496-2, 2001) and H.264 (ISO/IEC 14496-10, 2008), but each version shows an improvement over the

previous version. The MPEG 2 system (being primitive) does not have much added functionality to support diverse applications, but the MPEG 4 system has some additions that MPEG 2 does not have. The introduction of advanced error resilient tools, AC and DC prediction algorithms and object-based coding scenarios make MPEG 4 more suitable for a diverse range of applications. The latest standard H.264, which is termed Advanced Video Coding (AVC), is able to provide better quality video with high compression efficiency, with the aid of new variable length coding tools, such as CAVLC (Context Adaptive Variable Length Coding), Exp-Golumb and CABAC (Context Adaptive Binary Arithmetic Coding).

Each standard has its own profiles (versions) and corresponding functionalities. These profiles are classified based upon the capability and the requirements of the target decoder. The handheld device would be capable of handling only a simple profile, whereas the DVD players could handle advanced and high profiles. Table 2-1 lists the different profiles available in each MPEG standard.

Table 2-1. MPEG Family Profiles

MPEG 2	MPEG 4	H.264
Simple profile	Simple Profile	Baseline profile
Main profile	Advance simple profile	Main profile
4:2:2 profile	Advance real time simple profile	Extended profile
SNR profile	Main profile	
Spatial profile	Core profile	
High profile	Advance profile	

The problem of data loss occurring in a low bit rate wireless multimedia transmission is addressed in the second part of the thesis. The simple profile MPEG 4 codec is used for testing, as the codec standardised for 3G transmission by 3GPP (Third Generation Partnership Project) is similar to the MPEG 4 codec and has been assigned a unique extension of '.3gp'.

2.5. Wireless Fading Channels and Models

The different wireless standards in existence were reviewed in the introduction, so this chapter extends the discussion to the mobile channel characteristics. The mobile communication channels exhibit different error characteristics to traditional channels (Otani et al., 1981). The errors occur in the channels in bursts or clusters. The traditional AWGN (Additive White Gaussian Noise) models, which emulate single bit errors, no longer apply to the mobile channels. It becomes essential to understand the channel characteristics of the mobile channels and choose an appropriate model for the research. A short review of the error characteristics of the fading mobile channels is given and their behaviour is explained. In general, mobile channel characteristics can be classified into two categories: large scale and small scale propagation (Chryssomallis, 2002). Large scale propagation occurs when there is a line of sight between the transmitter and the receiver. The impact of large scale propagation on the received signal power is very gradual. Small scale propagation occurs when there is an abrupt change in the received power over short distances.

In a typical mobile environment, the transmitted signal will propagate in different paths and reach the receiver at different time instants with varying amplitudes and phases. This causes a phenomenon known as fading whose characteristics are unique for wireless channels. The fading normally occurs due to reflection, diffraction and scattering (Rappaport, 2005). It is observed that the large scale losses occur when the distance between the transmitter and the receiver is too large or they can be due to the improper placement of the transmitting antenna. These problems can be mitigated by proper planning and the correct placement of the base stations. Normally, modelling is undertaken using either the Longley-Rice (Longley and Rice, 1968) or the Okumura-Hata (Hata and Masaharu, 1993) model and the mobile network is designed in such a way that the path losses are minimised. The Longley-Rice model is mathematical in nature and the Okumura-Hata model is statistical in nature, as it utilises real time measurements.

The greater data loss occurs with the small scale fading, as the large scale fading can be easily predicted and resolved. The small scale fading occurs due to movement in the receiving mobile terminal and can be represented as a function of

velocity and the direction of movement. The rapid change in the position of the mobile terminal leads to constructive and destructive interference and results in rapid fluctuations of the signal (Chryssomallis, 2002). The fluctuation in signal power due to small scale fading can be represented by Rayleigh distribution and the fluctuation due to large scale fading can be represented by Rician distribution (Pätzold, 2002). The Rayleigh distribution becomes Rician if there is a dominant frequency component present in the signal, in other words, if there is a line of sight between transmitter and receiver. The signal falls below a certain level for a period of time and then reverts back to the normal state. This distribution makes the channel error characteristics in bit layer bursty or clustery in nature. There are formulae for predicting the average fade duration and the number of level crossings. The level crossing signifies the phenomenon of the received signal dropping below a particular threshold. Though the primary link layer parameters can be estimated using the formulae, they will not reflect the randomness of the channel; in other words, the location or occurrence of the bursty errors cannot be calculated using the formulae.

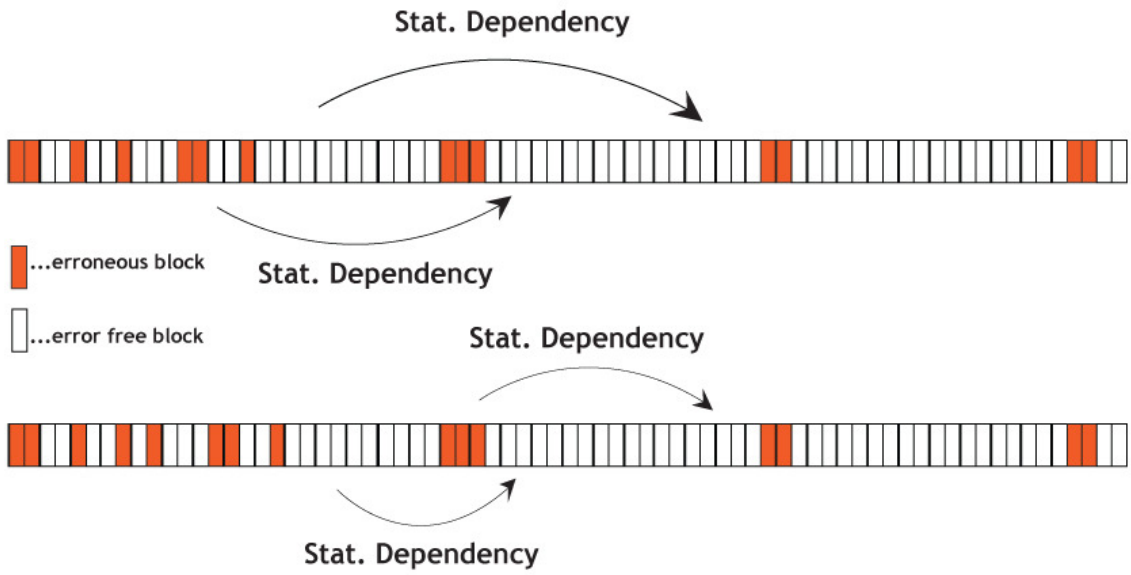


Figure 2-9. Bursty channel errors (Karner, 2007)

If a channel model does not reflect the random nature of the channel, then the test bench might not reliable enough to allow any conclusions to be drawn. The randomness in the channel models can be incorporated by using a stochastic process such as a Markov chain. Many models have been proposed over the years to portray

the channel realistically. The objective of the research is not to investigate the mathematical details of the models, but to choose the best model for the proposed research. The most popular model is the Gilbert channel model (Gilbert, 1960), in which the channel is modelled as a first order Markov channel. In a first order Markov chain, the current channel state depends only on the previous state. The states are modelled as good and bad states, where the channel is error prone in the bad state and errorless in the good state. The transition between good and bad states occurs at random; this makes the Gilbert model a hidden Markov chain. This model was modified in the Gilbert-Elliot model (Elliot, 1963), where the good state also carried some error probability. Though the Gilbert-Elliot model is as popular as the Gilbert model, both models received much criticism from researchers who were trying to model the wireless channel from real-time measurements (Karner, 2007). All the researchers agreed that bursty channel's errors were statistically dependent, but there were disagreements regarding the transition matrix and the length of the bursts. Figure 2-9 shows the statistical dependency exhibited by the bursty channel.

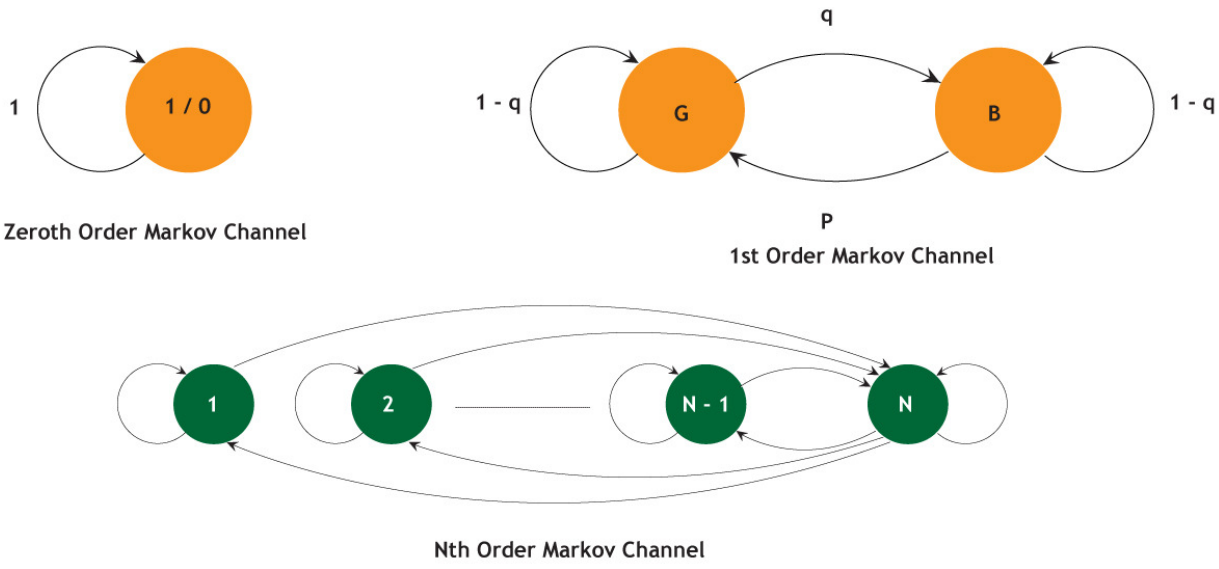


Figure 2-10. Stochastic channel models

Researchers like Fritchman (1967) used the Gilbert model as a base and increased the order of the Markov chain so that there were states where the errors occurred both in short and long bursts. The method proposed by Karner (2007) was similar to the one explained by Fritchman, but used the Weibull probability

distribution to control the transition between the states. There are many methods of channel modelling; since it is not an objective of the research to design a channel model, the most popular Gilbert method proposed in the MPEG reference software for simulating bursty errors is used for the research. A detailed description of the model is given in Chapter 9. Figure 2-10 shows the principle of operation of popular stochastic channel models.

Various channel coding schemes have been standardised to increase the error resilience of the transmission by adding redundancy to the data. Block coding, as the name signifies, processes the data into frames or blocks (Moreira and Farrell, 2006). The blocks of source bits are appended with some corrective bits, which can be processed in the receiver to retrieve the data bits in the event of errors. The convolutional code is a special kind of code, where the error correction bits are embedded into the bitstream through continuous processing (Johannesson and Zigangirov, 1999). Convolutional codes are widely accepted in mobile wireless systems due to their effective architecture and performance. The coding rate of the convolutional codes could be varied by a puncturing mechanism, which shows its flexibility towards adaptable applications. The latest addition to channel codes is 'Turbo code', which is widely used in most wideband mobile protocols due to its robust error correcting capabilities (Schlegel and Perez, 2004).

2.6. Summary

This chapter reviewed the background literature relating to the research. The chapter explained how the advent of a new standard in the broadcasting industry has triggered a new set of research problems due to compatibility issues. The review also showed that for a system to live through the change in customer requirements, it must be highly adaptable in embracing modifications rather than being a rigid standard. The chapter explained the video compression system and the complications involved in the coding process; for a system to operate a video codec in real-time, it must be equipped with reasonable hardware to handle complex processing. In addition, the methods, like variable length coding and motion prediction, impose a certain level of statistical dependency among the bits in the video stream. The fading characteristics of the mobile channels that are Rayleigh distributed result in bursty

errors, where the errors occur in clusters. This deviant behaviour of the error characteristics in the mobile channels from traditional Gaussian wireless channels, characterised by random errors, raises questions about the suitability of the channel codes for bursty errors.

In the next chapter, the errors occurring in the different layers of the protocol stack are explained in detail. It will be evident from the discussion in the next chapter how the convergence of the various technologies results in as many problems as it solves.

3. Video Editing and Transmission Layer Issues and Research Techniques

3.1. Introduction

From the reviews presented in Chapter 2, a view of the problems in this field was formed. In this chapter, the problem statements are presented for field reversal, mixed pulldown and channel error issues. A review of current methods used to solve the relevant problems is given along with the research techniques. Since part of the research is a commercial problem, relevant industrial terminologies are introduced in appropriate sections to illustrate the global nature of the problems. Section 3.2 presents the problem definition for the editing layer issues, which is followed by the problem definition for the transmission layer issues in section 3.3. The section after that explains the research techniques, followed by a summary in section 3.5.

3.2. Editing layer Errors

The problem of interlaced-progressive coexistence was solved by de-interlacing algorithms; however, two new problems then emerged in the commercial broadcasting industry. The issues are: -

- Field reversal
- Mixed pulldown

These problems are caused by the advent of digital technology and progressive displays, and were not apparent with the 'old technology' of mainly analogue displays. These problems were reported by commercial broadcasters to Tektronix as they were causing serious quality issues.

3.2.1. Field Reversal Issue

The first problem is known by several names and can be referred to as 'field dominance', 'field order error' or 'field swap'. Video streams are coded in such a

way that high fidelity is achieved in progressive displays. In third world countries, where usage of the progressive displays is less likely to penetrate the market in the near future, some serious artefacts are observed, while viewing the same video stream on traditional analogue CRT interlaced displays. The investigation into this problem was carried out in collaboration with Tektronix Plc for Google and MTV.

In field-based video content, which is designed to be reproduced on an interlaced display, there is a possibility of displaying the top and bottom fields in an incorrect order. Instead of displaying the fields in a strictly increasing temporal order, i.e., 1-2-3-4-5-6-7-8-9-10, the fields are displayed in an incorrect order, e.g., 2-1-4-3-6-5-8-7-10-9. The problem is given the name 'field reversal', which reflects the true nature of the problem. This results in the video frames not being displayed in a chronological order (Baylon and McKoen, 2006).

The field order of an interlaced video sequence is most well known by the term 'field dominance'. The field dominance of a video sequence is the order in which the fields of a frame should be displayed. The PAL and NTSC television standards are good examples of interlaced video (Murphy, 1989; Jack, 2004). In PAL video, the F1 field is the top field (the even line field followed by the odd line field) and for NTSC video, the F2 field is the top field (the odd line field followed by the even line field). The even line field is also termed the 'top field' and the odd line field is also termed the 'bottom field'. In other words, PAL exhibits top field dominance and NTSC exhibits bottom field dominance. Figure 3-1 shows a graphical illustration of the field reversal issue. In the figure, the capture format of the video is top field first (the even lines are captured first followed by the odd lines), which is PAL standard. An error in the bitstream results in the video being displayed bottom field first instead of top field first, which results in field reversal. If field reversal occurs in an interlaced frame, the edges and contours will be juddery (shaky) in appearance. Stationary portions of a field reversed video sequence will not display any visual disturbances, but any motion in the video would have a 'juddering and jittery (jumpy)' appearance as the motion flow is reversed.

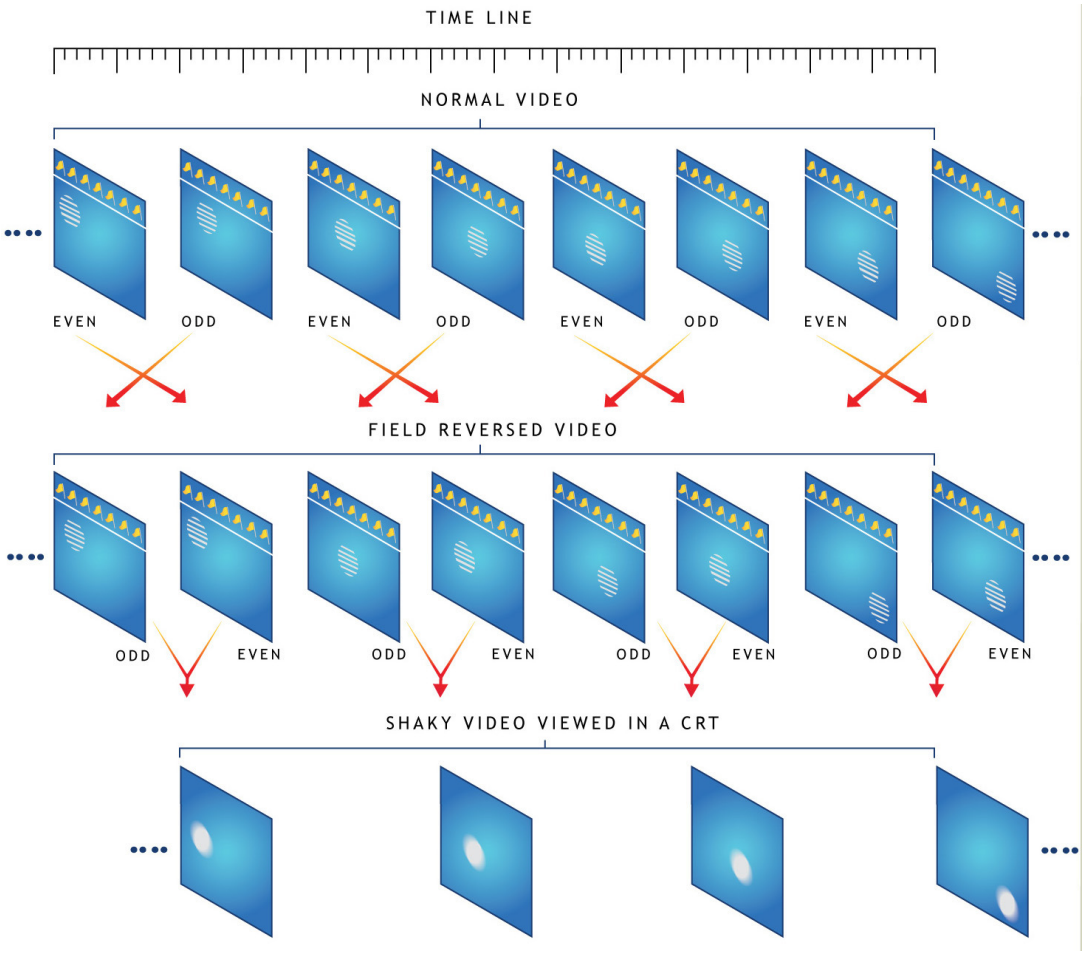


Figure 3-1. Field reversal error

A video is considered interlaced if the fields constituting a frame belong to two different time instants. A progressive video contains flat frames: the fields of a frame belong to the same time instant. An interlaced frame looks like a progressive frame, when the inter-field motion between two fields is negligible. Field reversal has no effect on a progressive frame; it does not make any difference which of the fields is displayed first, as successive fields are combined to form a frame before being displayed.

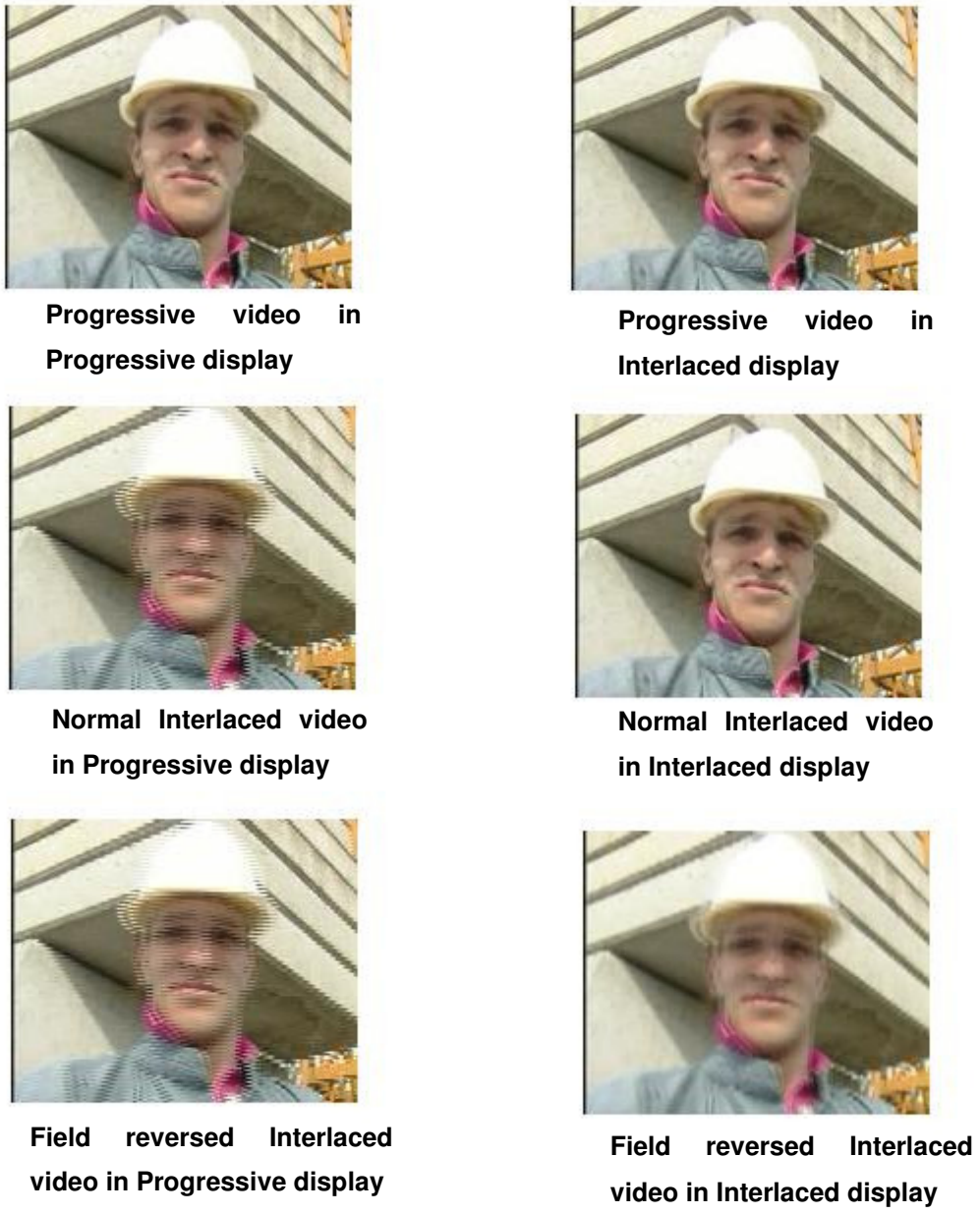


Figure 3-2. Impact of various video types on the displays

Figure 3-2 illustrates the impact of various video types on the interlaced and progressive displays. The first row shows that the subjective impact of a progressive video on both progressive and interlaced displays is the same. The second row shows that the interlaced videos exhibit combing artefacts in the progressive displays as both the fields are displayed at the same time instant, whereas the combing artefacts are not visible in the interlaced display due to the persistence of vision. The third row shows that the impact of field reversed video on a progressive screen is the same as

the normal interlaced video, whereas the interlaced display shows a juddered appearance as the motion trail from the future frame is visible in the current frame due to field reversal. In real time, the artefact is much worse, as the artefact occurs in the temporal domain, not in the spatial domain.

Table 3-1. Flags in metadata across various standards

Standards	Flags
DV-25	FF: Frame/field flag FS: First/Second field flag FC: Frame change flag IL: Interlace flag
VC-1	INTERLACE: Interlace content FCM: Frame coding mode PSF: Progressive segmented frame TFF: Top field first RFF: Repeat first field PULLDOWN: Pull down flag
H.264	FIELD_PIC_FLAG BOTTOM_FIELD_FLAG PIC_STRUCT
MPEG-2 / MPEG-4	PROGRESSIVE_SEQUENCE VOP_STRUCTURE/ PICTURE_STRUCTURE PROGRESSIVE_FRAME TOP_FIELD_FIRST REPEAT_FIRST_FIELD

This field dominance should be constant throughout the content, but this does not normally happen in real time, as the video goes through a long and repetitive editing process before being broadcast. Video content of different standards, types and qualities is sent to the 'editing factory' (post production); the edits are made based on the specifications of the broadcasters and transcoded into a common standard (Gardner and Scoggins, 1991). Each video content has its own protocol; when the video clips of different protocols are merged, the fundamental field dominance information encoded in the metadata is missed out on most occasions, which in turn, leads to field reversal. Table 3-1 lists the flags used to signal the field order across the video standards. Detailed information on the flags is presented in Appendix A.

A flag is present in the metadata of most encoded video streams; the flag informs the decoder which field is to be displayed first. If this flag does not match with the interlaced source material then field reversal error occurs. If videos with different field dominance are edited together, the information about the change in field dominance should be clearly signalled in the metadata of the final stream during transcoding, but most of the time this is not done.

For example, consider interlaced material in NTSC format that is edited with PAL material and transcoded into MPEG-2 format. In MPEG-2, the *Top_Field_First* flag is used for defining the field order of a frame. The section of the hybrid video sequence that contains the video frames from PAL material must have the *Top_Field_First* flag set to '1', as the frames must be displayed top field first followed by bottom field. Similarly, the section of the hybrid video sequence that contains the video frames from the NTSC material must have the *Top_Field_First* flag set to '0', as the frames must be displayed bottom field first followed by top field. If this editing procedure is not followed, field reversal occurs. This happens due to the use of unskilled workers in the editing factory, who have very little knowledge about the standards (Anderson, 1999). Most editing factories use the progressive display to check the quality of the video after editing; this is manually performed by 'eyeballing' (quality control by manual viewing), an industrial term for visual inspection. A field-reversed interlaced video displayed in a progressive display will not show any visual artefacts, as the fields captured at

different time instants are displayed in the same time instant (that is, 50 fields/sec video is displayed as 25 frames/sec video). The video is approved as having passed the quality check and is sent to the broadcasters after a visual inspection.

The field reversal error destroys the integrity of the bitstream and continues to corrupt the videos with which it is edited, as the essential field dominance information has not been passed on. The essential information is lost, and the mistake cannot be identified and rectified unless the whole stream is played on a CRT and visually inspected. According to information from the broadcasters acquired by email and telephone conversations, there are numerous clips already in the market that have field reversal errors, which makes the process of the retrieving all the clips to the editing houses and performing a visual inspection on the streams impossible.

3.2.2. Mixed Pulldown Issue

The second problem comes under the telecine domain, which is a process of digitising film material. The root cause of this problem is the reverse of the scenarios that cause field reversal. The roots of this problem lie with the methods designed from the perspective of seeing interlaced video as the only future standard. This thesis reports on an investigation that was carried out in collaboration with Tektronix Plc for Microsoft on mixed pulldown.

Pulldown is the process of increasing the frame rate by adding redundant fields, whereas reverse telecine is the process of decreasing the frame rate by removing the redundant fields. It is easier to perform reverse telecine if the pulldown pattern and location of the redundant fields are known beforehand. Discontinuities can occur in the pulldown pattern if edits or effects are applied to telecine content and this can cause errors when attempts are made to reverse the telecine content (Hui, 2005; Keating and Richards, 1995).

Since reverse telecined content is designed to be reproduced on a progressive display, there should be no interlacing artefacts (combing) after the reverse telecine process, because the original material was frame-based. The presence of combing artefacts in a reverse telecined video will indicate that the reverse telecine process

had been performed erroneously (Coombs et al., 1996). The pulldown frame is not a physical frame, but a virtual frame, so the presence of a pulldown frame should be signalled in the bitstream with a special flag. The flags that convey the redundant field information are given in Table 3-1 along with the flags that convey field order information. When the decoder reads this flag, it understands that the field is to be displayed for longer than the average display period, which in turn increases the frame rate as required. While performing reverse telecine, software can run through the bitstream and reset the flags to restore the up-sampled frame rate to 24 frames/sec. Another significant advantage of signalling the repeat fields rather than performing the physical insertion is that the pulldown frames exhibit combing artefacts, as they constitute fields from different time instants. This results in an increased consumption of bits in comparison to a normal frame due to increased spatial frequency.

The flags in the metadata are ignored most of the time while frame rate of the video sequence is increased using the using pulldown frames. This results in pulldown frames being present as physical frames in the video stream, which in turn reduces the compression efficiency. If the video stream is to be reverted to the original 24 frames/sec sequence, the inverse telecine circuitry must be used to remove the redundant frames. The inverse telecine operates in the image domain and looks for a fixed number of redundant fields every ‘n’ frames using correlation methods.

If a pulldown stream is edited with another pulldown stream, the editing must be performed carefully so that there is no disruption in the pulldown pattern. In Figure 3.3, two pulldown video sequences are edited together. The pattern repeats every five frames; the correct process of editing will be to edit the streams at a frame number that is a multiple of five, so that the pattern carries on, without resulting in an error while performing reverse telecine. Instead, the video stream has been edited at the incorrect location. While performing inverse telecine, incorrect frames are removed from the sequence due to the occurrence of redundant frames in the wrong location. This results in a frame rate down-converted video sequence with pulldown frames still present in the stream. If the flags are not set for a pulldown frame in a video bitstream, the virtual frame becomes a physical frame and the true information

on the video frame is lost for ever: “A vast number of films have already been transferred to the video without these special flag signals, and will not be practical to return to the original film sources in most cases” (Coombs et al., 1996).

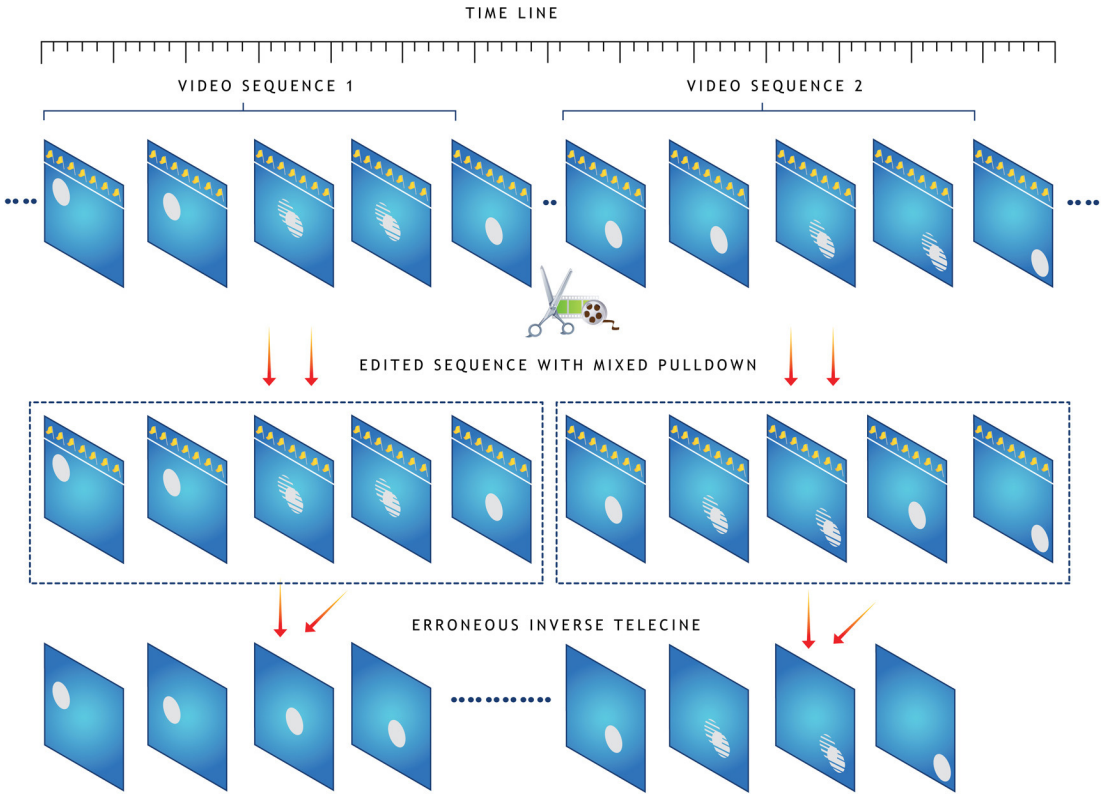


Figure 3-3. Mixed pulldown error

A video stream with a mixed pulldown error will corrupt the videos with which it is edited and propagate the error on a very large scale. The solution to the problem is to identify the video frames that have redundant fields and signal the flag in the bitstream, which will serve as very valuable information while editing operations are being performed on the stream. Since the problem is more complicated with pulldown frames with a non-uniform pattern, the traditional methods cannot be used for detecting the pulldown frames, as they may yield false positives (Christopher and Correa, 1997). The check must be performed on a frame-by-frame basis to choose the pulldown frames rather than using windowing methods (correlation over multiple frames).

3.3. Transmission Layer Errors

The channel errors damage the data integrity whilst in transmission. When the video data is transmitted over wireless networks, the quality of the reconstructed video has a random relation with the bit errors occurring in the lower layers. This is due to the use of the variable length coding method in compressing the video frames. One bit error in the wrong location can initiate error propagation and affect the quality of the succeeding frames despite there being no other errors in the bitstream (Yoo, 1998). This phenomenon is known as 'propagation losses'.

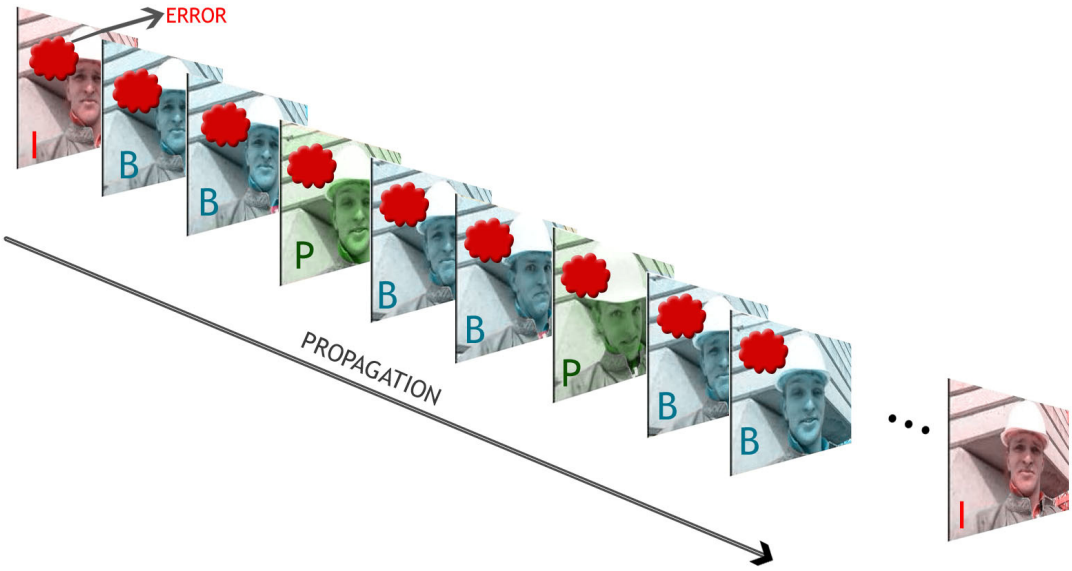


Figure 3-4. Error propagation in a video stream

The predictive nature of the P and B frames makes them dependent on the previous frame for the accurate reconstruction of the current frame. This is because the motion vectors and the residual data are derived by prediction from the previous frame. The motion vector describes the location of the block in the previous frame with which the block in the current frame is highly correlated and residual data are the difference values between the blocks. So when a block in the previous frame is lost due to an error, this results in all the blocks that depend on the lost block for reference being erroneous. These corrupted blocks would have served as a reference to other blocks in the future frames; a chain reaction is initiated leading to an avalanche effect on the quality of the reconstructed video. This is known as 'error

propagation' or 'propagation losses' in a video coding context. This is graphically illustrated in Figure 3-4.

This issue of 'propagation loss' has been widely accepted as a problem and there has been considerable research in this field. A broad literature review on existing methods is undertaken in the following sections, as there are multiple directions from which to approach the problem. The methods in existence can be broadly classified into three categories:

- UEP (Unequal Error Protection) and Joint-Source-Channel Coding (JSCC)
- Cross-layer design models
- Source error resilience

The UEP is the process of partitioning the bitstream into segments of varying importance, and subsequently, the segment with high importance is offered a high level of protection, whereas the segment with low importance is offered a lower level of protection. The Joint-Source-Channel coding is the process of applying UEP without any significant increase in the bitrate (the rate matching source coding bits against channel coding bits). Some methods proposed under this category place more emphasis on the application of variable channel codes to different partitions than on the mechanism of partition (Srinivasan et al., 2004; Barmada et al., 2005), whereas other methods place more emphasis on the method of partition than on the method of channel coding (Qu et al., 2004; Yan and Ng, 2003). Some methods place extra emphasis on the metrics used in partitioning the video (Reibman et al., 2001; Bouazizi and Gunes, 2004).

The second common method in use is based on the principle of 'cross-layer' design. Cross-layer design models rely on the feedback from the channel about error rates and dynamically vary the transmission signal characteristics in accordance with the channel feedback (Ma et al., 2005; Alexiou and Bouras, 2005; Kodikara et al., 2004). The feedback parameters include signal to noise ratio, block error ratio, bit error ratio, transmit power, bit energy and delay.

The third method is based on the source-based error protection, where the design of the methods is confined within the source coding domain and the methods do not interfere with other layers of the protocol stack. The methodology includes interleaving, intra-blocks insertion, variable fragmentation, retransmission, missing motion vector estimation, fixed length coding, spatial and temporal concealment (Xu and Zhou, 2004; Ghanbari and Bober, 2002; Dogan et al., 2002).

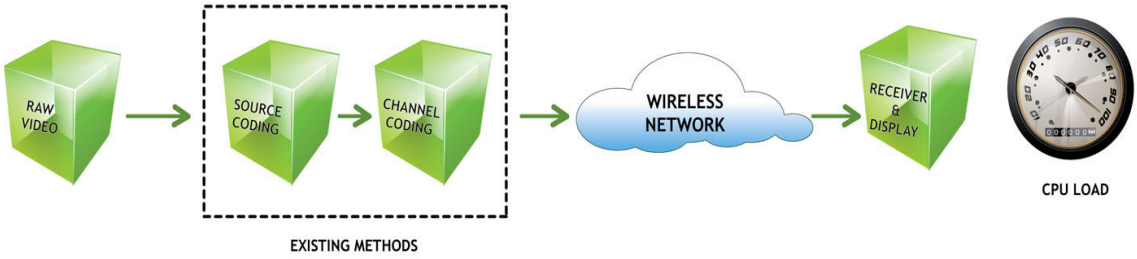


Figure 3-5. Error protection procedures in existence

Despite the considerable amount of research that has been conducted addressing this problem, all the methods have a common disadvantage: none of the methods are realistic. The methods claim performance improvement in terms of quality and other objective factors, but their implementation in real-time is still very distant for the following reasons: -

- It is not easy to change the rules of the transmission on a mobile protocol stack, as it is standardised for global usage. The methods must have this as the primary constraint. The UEP, JSCC and cross layer models require a change in the way the data are transmitted across a standardised protocol stack, which is very unlikely to happen. This makes the methods unsuitable for real time implementation. Methods might show excellent performance, but if the reality factor is not embedded in the algorithms, the methods cannot progress past the simulation stage to real-time implementation.

- The channel error coding methods were primarily designed for basic data communications. The channel coding operates on the lowest layer

of the protocol stack; in other words, it is independent of the nature of the application. The code rate can be increased or decreased to improve or reduce the error correction capability of the channel codes. The importance of a particular bit in the video stream cannot be understood from the channel coding layer; the cross layer design methods must be used, which will make the methods unrealistic for practical implementation.

- The methods classified as source error resilient methods restrict their design process within the source coding domain, which makes them more suitable for real-time implementation. However, it is clearly evident from the discussion that most of the algorithms are decoder-centric. The error concealment happens in the decoder, which increases the decoder's complexity. Since most terminals receiving data across the wireless mobile channels will be battery operated, this will increase the power consumption. In addition, if the architecture of the error concealment process is decoder-centric, it must be fast enough to operate at high refresh rates; this will result in performance variation across mobile terminals with different hardware capabilities. Figure 3-5 shows the operating methodology of existing methods.

3.4. Research Techniques

This section explains a very broad techniques used in the research, because the research branches out to many sub-domains that have techniques of their own. The techniques of the sub-domains are explained in the relevant sections of the thesis. Since the algorithms are intended for direct implementation, care has been taken in designing the techniques, so that none of the existing standards or protocols is infringed.

3.4.1. Research Techniques for Editing Layer Issues

The editing layer issues can be resolved by visually inspecting the video stream; however, since this is a tedious process, it is very unlikely that the

postproduction houses will follow this method. The solution to the problem is to develop an algorithm that can emulate the visual inspection process (Meng and Chang, 1997). Since the algorithm must operate on the image data, the appropriate location for the algorithm is the source coding domain. The algorithm will restore the integrity of the clips being sent to the editing houses. There are numerous clips currently in the market that have field reversal issues, but it would be impossible to identify the clips and return them to the editing houses for processing. The practical solution to the problem is to upgrade the software of the terminal equipments, so that the frames can be displayed without artefacts. Digital televisions and setup boxes are designed with the ability to upgrade their software (Pellegrini et al., 2008; Liu, 2007). The algorithm designed must be robust and fast enough to operate on the video frames on-the-fly with good precision so that they could be implemented in the source decoding end.

In the first part of the thesis, the primary problem to be solved before addressing field reversal and mixed pulldown problems is to estimate the inter-field motion to identify the video frame type. Unless the scanning mode or the source of a video frame is known, it is not possible to apply higher layer algorithms to rectify the errors. For example, if a frame is detected as interlaced, then the field reversal algorithm must be applied to verify the field order; if a frame is detected as pulldown, then the mixed pulldown algorithm must be applied to verify the presence of a repeat field. Alternatively, if a frame is detected as progressive, it will not produce any visual artefacts, hence there is no need for any processing, but the application of higher layer algorithms will result in false positives.

There are many methods that could be used to detect the visual combing artefacts, but they lack robustness and consistency across images of different resolution, spatial detail and quality. Further, there is vertical phase lag between two fields, which also contributes to inconsistency (Hui et al., 2000). Most methods use the difference in pixel values to estimate the inter-field motion. A common threshold can never be set for the classification. Threshold values are generally made adaptable or set to a very safe value to avoid false positives; thus, the adaptive process is subject to failure.

The review of de-interlacing methods suggests that there is no need for an effective inter-field quantifier (De Haan and Bellers, 1998). The miscalculation of the motion between the fields, at worst, will lead to extra smoothing of static areas of the picture. This quality degradation is negligible while viewing a video at a refresh rate of 30 frames/sec; however, if there is miscalculation of motion information while applying field reversal and mixed pulldown algorithms, it will result in false positives.

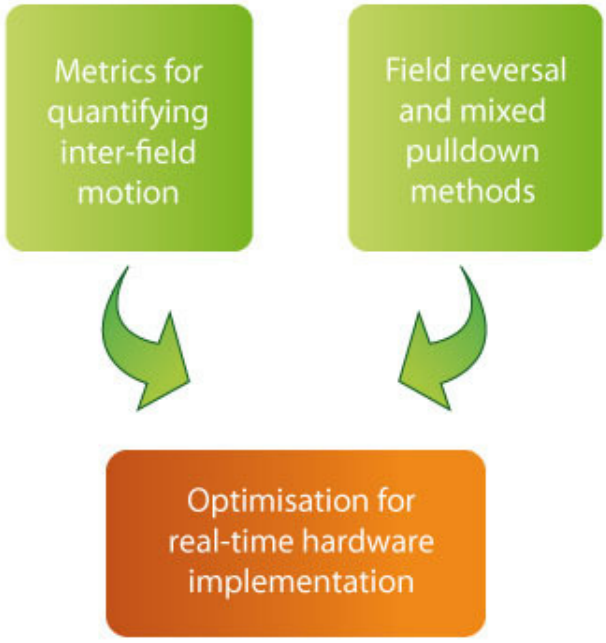


Figure 3-6. Approach of research for editing errors

In Chapter 4, the design of a novel and precise metric that can capture even meagre inter-field motion and quantify it within a constant dynamic range is described. The drawbacks of the existing methods are carefully analysed and the inter-field quantifier is designed to have a common threshold regardless of varying image characteristics.

Once an inter-field quantifier has been designed, the field reversal and mixed pulldown algorithms can be implemented with good precision as the inter-field quantifier will capture even meagre motion drift between the fields and visualise the level of impact of the artefacts on the observer. The solution for field reversal and

mixed pulldown lies in estimating the continuity of motion among the video frames, as field reversal causes the frame to be displayed in the wrong order, and the mixed pulldown results in motion aliasing due to the redundant fields.

In Chapter 5, a suitable correlation method is designed for detecting field reversal and mixed pulldown errors. A popular correlation method is chosen to measure the motion flow along the frames of the video sequence, and subsequently, real time constraints are embedded in the design process to shape it for commercial implementation. With the proposed methods in place, the decoded video can be subjected to image analysis and the results can be verified automatically against the metadata present in the video bitstream. In the event of a field reversal error, the parity of the *Display_Order* flag is changed to reflect the results of the baseband image analysis. In the event of the mixed pulldown error, either the frames are removed or the *Repeat_Field* flag is set in order to restore the integrity of the bitstream. Figure 3-6 shows the block diagram of the approach used in addressing the editing layer issues.

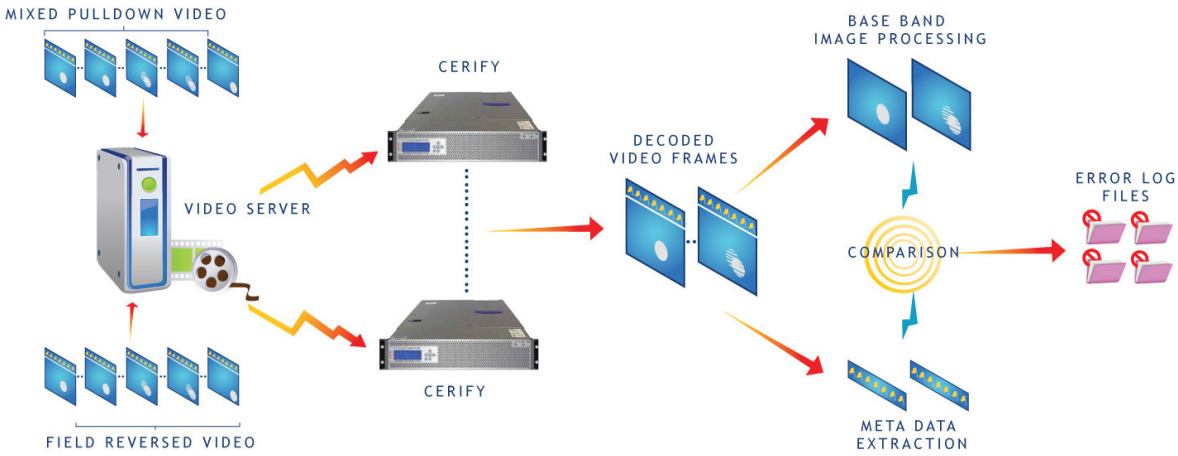


Figure 3-7. End-to-End architecture of Tektronix Cerify testing system

The algorithms are designed to be implemented commercially in the Tektronix equipment 'Cerify', which is a real time video quality control system. Designing a complicated image processing algorithm to be implemented on a real time system posed many limitations, as the speed of the algorithm was the primary constraint. In Chapter 6, the simulation results are presented. Two test benches are used to test the

design; the initial prototype is designed in Matlab, and later the codes are rewritten in C++ for integration into the Tektronix 'Cerify' system. Figure 3-7 shows the graphical illustration of final architecture of the Tektronix Cerify kit with the proposed methods integrated in it.

3.4.2. Research Techniques for Transmission Layer Issues

In the second part of the thesis, the approach of the research is carefully designed in such a way that the drawbacks of the existing methods for counteracting channel errors are not repeated (Wang and Zhu, 1998). The designed methods are confined within a layer of the protocol stack for ease of integration with the existing architecture, and will not incorporate any cross-layer design methods. Since the domain that is chosen must have enough flexibility and must allow modifications so that it can be incorporated without disturbing other protocols, the source coding domain was chosen (Van der Schaar et al., 2003; Carle and Biersack, 1997). Any change that is made in the application layer (source coding) can be applied in real-time by sending a simple update to the decoder module to the terminal (Rungta et al., 2009). This is not true with other layers of the protocol stack (Tangemann and Rheinschmitt, 1994). The block diagram of error resilience confined in the source coding block is shown in Figure 3-8.

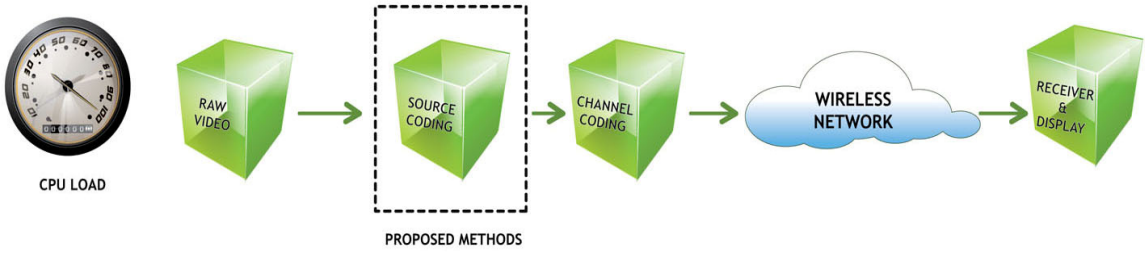


Figure 3-8. Source based error resilience

In Chapter 7, the appropriateness of the choice of the source coding layer is justified by comparing the performance of the end-to-end mobile multimedia system by moving the error concealment block across different layers. The investigation starts with a discussion and an experiment to prove that the traditional channel codes may not be suitable for providing error resilience to mobile multimedia

communications, as the application is highly data dependent and the errors are bursty in nature. Despite there having been a considerable amount of research into this field over the years with different methodologies, the protocol stack is not adapted to use the methods. The chapter also argues that if previous research had been directed towards the source coding domain rather than relying on the channel coding, a more implementable solution could have been achieved. Some data hiding methods are explained to support the above discussion and to illustrate the flexibility offered by the source coding domain, which is absent in the channel coding domain.

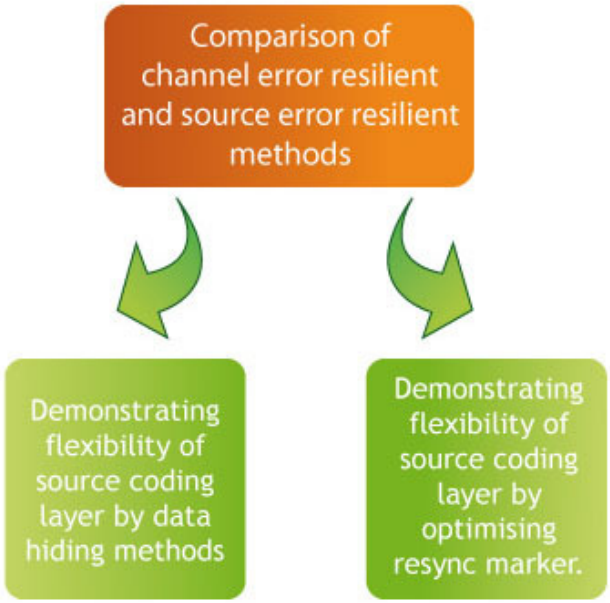


Figure 3-9. Approach of research for channel errors

The algorithms must be encoder-centric rather than decoder-centric; the burden of error protection should be shifted from the decoder to the encoder. The importance of designing the decoder as a 'dumb' terminal is explained by Girod (1999). In most cases, the encoder will be a video server that encodes the data and the decoder will be software located in the battery operated terminal with the exception of video conferencing applications, in which both ends will be battery operated. By designing encoder-centric methods, the decoder complexity of the battery operated terminal is decreased, but in turn, there is an increase in the encoder complexity of the video server, which is power operated. This results in an error concealment process that is simple, fast and efficient, which increases the battery life of the mobile terminal.

In Chapter 8, a specific source coding tool is chosen and optimised using encoder centric architecture to demonstrate how operating in the source coding layer enables a meaningful approach to the design process. The 'resync marker', which is primarily used by the decoder to regain synchronisation to the bitstream in the event of errors by addition of redundant bits, is investigated in detail. A smart novel encoder centric design is proposed from the source coding layer, so that, in the event of errors, the decoder will gain synchronisation by use of a mechanism that does not demand any physical bits in the bitstream.

The simulation results are presented in Chapter 9. The test bench for the transmission layer issue is in Matlab. The simple profile MPEG-4 video codec is used. The bursty channels are simulated using the Gilbert function described in the MPEG-4 reference software and the bit error ratio is set to 10^{-2}. The channel codes used are turbo and convolutional codes with the generator polynomials described in the WCDMA standard. Figure 3-9 shows the block diagram of the approach used to address the transmission layer issues.

3.5. Summary

The problem definition and the research techniques were presented in this chapter. Though the problems defined occur in different layers, the implementation of solutions to the problems will occur in the source coding domain. This will contribute to the broad objective of justifying the robustness of the source coding domain in concealing errors regardless of their location in the protocol stack. The field reversal error explained in section 3.2.1 was a problem that was unknown to the broadcasting industry, as each broadcaster reported the problem with a different name leading to ambiguity in understanding the cause of the problem. The work undertaken in this thesis has resulted in the global name 'field reversal' being given to the problem and has led to the generation of documents detailing the cause, implications and solutions in a form understandable by commercial broadcasters. The investigation of field reversal and mixed pulldown issues in a commercial environment has resulted in the critical analysis of methods designed for channel errors for real time implementation. As a result, none of the designed algorithms compromise parameters that are a primary requisite for real time implementation.

The transmission layer issue, which has been an active area of research for years, is still not saturated, in spite of dramatic advancements in wireless technology. This justifies the fact that any new algorithm, despite showing great performance improvement, must not infringe a standardised protocol stack, as it will not be possible to implement it in real time.

4. Quantifying Inter-Field Motion for Interlaced/Progressive Classification

When a raw video stream is transcoded into a compressed video stream for the first time, the metadata in the compressed stream accurately reflects the nature of the video frames in the video stream. As the video is edited with other streams of different origins, the metadata information is modified to reflect the characteristics of the majority of the video frames in the sequence, and as a result, the information on individual frames is lost. When the higher layer algorithms designed for processing video streams of a specific origin are applied to hybrid video sequences, it results in false positives. If the process of identifying the origin of a frame in a video stream using visual inspection is automated by an efficient computer vision algorithm, the accuracy of the higher layer algorithms can be improved. Designing a generic algorithm that is consistent across videos of unpredictable spatial and temporal characteristics is a challenging task. This chapter proposes methods for automatically identifying the origin of the video frames using advanced image processing principles. The contribution of the proposed methods to the precision of the higher layer algorithms is demonstrated in the next chapter.

4.1. Introduction

In commercial hybrid videos, the interlaced, progressive and pulldown video frames coexist in the same stream. Their characteristics can be identified only by the presence of the combing artefacts due to the inter-field motion. The inter-field motion is the motion drift between two successive interlaced fields captured at different time instants. The performance of the popular methods, such as de-interlacing and inverse telecine applied to television signals, depend on the inter-field motion. This requires the design of a unique inter-field quantifier that can provide a global metric to quantify the inter-field motion. There is no unique inter-field quantifier in existence because any designed method will require an adaptive threshold due to the varying spatial and temporal details of the frames in the video

sequence. In this chapter, a precise inter-field quantifier is proposed that uses advanced object segmentation and representation methods supported by some basic mathematical theories. Every layer of the algorithm is crafted with the intention of providing a metric that is unique and consistent across different standards, resolutions and quality by providing a stable threshold cut-off and dynamic range. The results show good performance in terms of consistency and the stability of the quantifier. As it is a requirement for field reversal and mixed pulldown algorithms to capture every motion drift between fields, however small, the inter-field quantifier will play an important role in increasing the speed and the performance of the algorithms by acting as a pre-processing module.

4.2. Review of Literature, Problem Definition and Chapter Organisation

Conventional methods look for the presence of combing artefacts in a frame for classification (Winger and Jia, 2006). This section investigates the current methods for interlaced/progressive classification and their drawbacks. In addition, the need for a unique robust metric and its impact on the reliability of the higher layer algorithms is explained. Most current methods for detecting the inter-field motion subtract adjacent fields and utilise the magnitude of the difference values for classification. The fields not only differ in temporal detail, but differ in spatial detail as well, as there is a phase lag in the vertical direction between two fields. The difference values are influenced by vertical phase lag and transmission noise (Christopher and Correa, 1997). This ambiguity has contributed to the lack of robustness in conventional methods for detecting inter-field motion (Winger and Jia, 2006).

The primary challenge of the research is to design a metric with a stable independent threshold cut-off for interlaced/progressive classification. Since the processing domain requires pixel difference values, the absolute sum of the resulting intensity values varies with the image area, spatial detail and noise. This results in an unstable threshold cut-off. If the threshold cut-off is adaptive, then the metric becomes statistically dependent on other parameters of the video stream, which in turn will affect the robustness of the metric. There are many methods in existence for interlaced/progressive classification and they suffer from drawbacks similar to those

explained above, so the methods are not consistent. Ozgen and Lim (2006) used SAD (Sum of Absolute Difference) values for inter-field motion estimation. Hui (2005) utilised pixel-by-pixel operations, whereas Conklin (2006) used both horizontal and vertical pixel difference values for the estimation. Winger and Jia (2006) used statistical information on a series of frames to improve the accuracy of the classification. The methods reviewed below are different from their traditional counterparts in a unique way. Baylon and McKoen (2006) used a linear invariant step invariant zipper filter to quantify the inter-field difference; this process is analogous to vertical gradient calculation. Martin and Smith (1995) used the ratio between the sum and the difference of the fields for interlaced-progressive classification.

The higher layer algorithms like de-interlacing, field reversal detection and mixed pulldown detection follow a flexible approach; some algorithms operate on blocks and some prefer frame-based processing. The architecture of the higher layer algorithms totally depends on the terminal hardware restrictions. At some point, the algorithms rely on the lower layer interlaced/progressive classification. In most cases, the existing algorithms have their own in-built module for inter-field motion measurement. This has led to all the algorithms currently in existence being application-centric, and has resulted in the lack of a global metric for interlaced/progressive classification.

This chapter offers some efficient solutions to the above-mentioned problems by proposing three robust metrics, namely, ‘convergence ratio’, ‘gradient deviation ratio’ and ‘cluster ratio’, to quantify the inter-field motion. A number of image-processing techniques are used to find an efficient solution for the problem of interest. Since the problem is of a commercial nature, emulation methods are used throughout the thesis. Emulation is the process of imitating something for a virtual presence. In this chapter, the simple image processing modules are put together in such a way that they emulate very highly complicated image-processing operations, for ease of hardware implementation.

This chapter starts with the frequency domain analysis of the interlaced video signals in section 4.3 to gain a mathematical understanding of the problem. In

section 4.4, the issue of handling fields that are not spatially equivalent is discussed with some mathematical proofs. Subsequently, two interpolation methods are chosen and transformed into frequency domain and some observations are presented. In section 4.5, the structure of the low and high frequency components resulting from the subtraction of the interpolated fields is explained. In section 4.6, novel object segmentation and representation methods are proposed to quantify the inter-field motion in line with the popular 'circularity' metric. The metric's relation to HVS (Human Vision System) is illustrated by translating the image space into visibility thresholds based on the spatial and temporal frequency bands. Finally, section 4.7 concludes the chapter by summarising the outcomes of the investigation.

4.3. Frequency Domain Analysis of Interlaced Signals

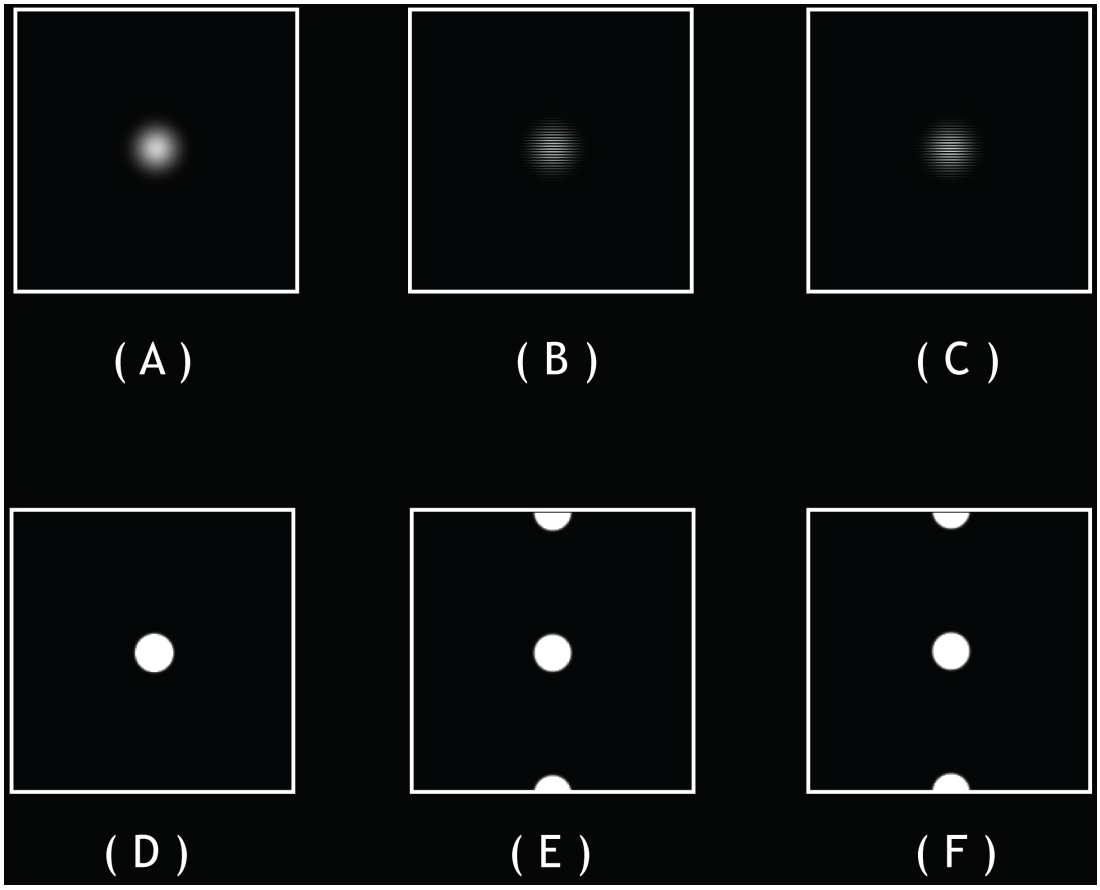
The seminal work by (Beuker and Shah, 1994; Shah and Beuker, 1993) on frequency domain analysis of the interlaced video signals is reviewed. Beuker and Shah used the Gaussian signal to explain the frequency domain concepts. The Gaussian signal is a very reliable signal for experimentation due to its spectral properties, as the Fourier transform of a Gaussian signal is Gaussian as well. An interlaced video frame consists of two fields captured at two different time instants whose frequency spectrum can be represented by equation (4-3-1), where 'F' is an interlaced video frame, 'F_E' and 'F_O' are extracted odd and even line frames, w_x and w_y are coordinates in the frequency plane expressed in radians relative to the sampling frequency.

$$F\left(w_x, w_y\right) = F_E\left(w_x, w_y\right) + F_O\left(w_x, w_y\right) \tag{4-3-1}$$

Extracting odd and even lines from the base band signal is equivalent to sub-sampling the signal by a factor of two. This sub-sampling process induces an alias term along with the base band term. The alias term present in two fields (4-3-2) (4-3-3) is of a different polarity as there is a phase lag in vertical direction between the fields.

$$F_E\ (w_x, w_y) = \frac{1}{2} F(w_x, w_y) + \frac{1}{2} F(w_x, w_y + \Pi) \quad (4\text{-}3\text{-}2)$$

$$F_O\ (w_x, w_y) = \frac{1}{2} F(w_x, w_y) - \frac{1}{2} F(w_x, w_y + \Pi) \quad (4\text{-}3\text{-}3)$$



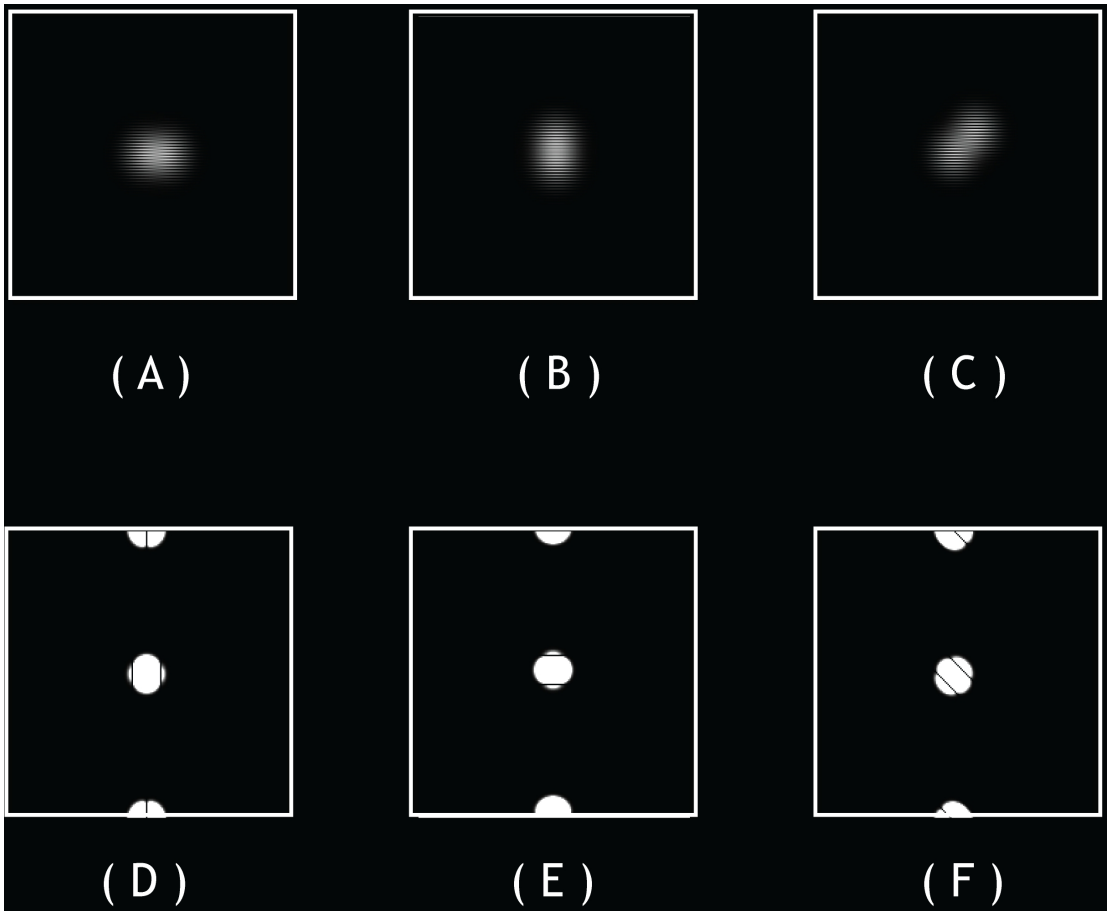
A. Gaussian Signal **B. Odd Line Field** **C. Even Line Field**
D. Gaussian Spectrum **E. Odd Line Spectrum** **F. Even Line Spectrum**

Figure 4-1. Interlaced spectrum of a static Gaussian signal

Figure 4-1 shows the frequency spectrum of a static Gaussian signal (Fig 4-1C, 4-1F), an extracted odd line frame (Fig 4-1C, 4-1F) and an even line frame (Fig 4-1C, 4-1F). The above equations are valid if there is no inter-field motion. In reality, there is some motion detected between successive fields, most of the time. Equation (4-3-4) shows the result of shifting a signal by ‘n’ and ‘m’ in horizontal and vertical

directions respectively. The shifting property of the Fourier transform is used in the right hand side of equation (4-3-4).

$$F\left(w_x + n, w_y + m\right) = F\left(w_x, w_y\right) e^{j(w_x n + w_y m)} \qquad (4\text{-}3\text{-}4)$$



A. Horizontal Motion **B. Vertical Motion** **C. Diagonal Motion**
D. Horizontal Spectrum **E. Vertical Spectrum** **F. Diagonal Spectrum**

Figure 4-2. Interlaced spectrum of a moving Gaussian signal

This section illustrates how the motion modifies the interlaced frequency spectrum. The spectrum of the even field (4-3-5) remains the same, as only relative motion between two fields is taken into consideration. The spectrum of the odd field is multiplied by the exponential motion coefficients (4-3-6) based on the shifting principle illustrated in equation (4-3-4). F_M is an interlaced video frame in which there is some motion between the fields; 'F_{EM}' and 'F_{OM}' are extracted odd and even lines respectively.

$$F_{EM}\ \left(w_x\ ,\ w_y\ \right) = \frac{1}{2}\ F\left(w_x\ ,\ w_y\ \right) + \frac{1}{2}\ F\left(w_x\ ,\ w_y + \Pi\ \right) \qquad (4\text{-}3\text{-}5)$$

$$F_{OM}\ \left(w_x\ ,\ w_y\ \right) = \frac{1}{2}\ F\left(w_x\ ,\ w_y\ \right) e^{j(w_x n + w_y m)} - \frac{1}{2}\ F\left(w_x\ ,\ w_y + \Pi\ \right) e^{j(w_x n + (w_y + \Pi) m)} \qquad (4\text{-}3\text{-}6)$$

The resultant video signal (4-3-7) is the summation of odd (4-3-5) and even (4-3-6) signals. From equation (4-3-8), it can be observed that the base band signal and alias signals are multiplied by similar components with different signs.

$$F_M\left(w_x\ ,\ w_y\ \right) = F_{EM}\ \left(w_x\ ,\ w_y\ \right) + F_{OM}\ \left(w_x\ ,\ w_y\ \right) \qquad (4\text{-}3\text{-}7)$$

$$F_M\left(w_x\ ,\ w_y\ \right) = \frac{1}{2}\ F\left(w_x\ ,\ w_y\ \right)\left[1 + e^{j(w_x n + w_y m)}\ \right] - \frac{1}{2}\ F\left(w_x\ ,\ w_y + \Pi\ \right)\left[e^{j(w_x n + (w_y + \Pi) m)} - 1\ \right] \qquad (4\text{-}3\text{-}8)$$

$$F\ \left(w_x\ ,\ w_y\ \right) = F_M\left(w_x\ ,\ w_y\ \right) \qquad If\ m = n = 0 \qquad (4\text{-}3\text{-}9)$$

If there is no motion, equation (4-3-8) reverts to equation (4-3-1). The frequency spectrums of the Gaussian signal with horizontal, vertical and diagonal motion are shown in Figure 4-2. The horizontal motion generates vertical lines in the spectrum (Fig 4-2A, 4-2D), the vertical motion generates horizontal lines in the spectrum (Fig 4-2B, 4-2E), and the diagonal motion generates diagonal lines in the spectrum (Fig 4-2C, 4-2F). The lines get closer with an increase in motion between the fields. The lines in the base band component and the alias component differ by a phase of $\pi/2$.

These results can be used to understand the drawbacks of existing methods and to identify what is required of an algorithm for it to offer consistency and robustness across images of varying characteristics. For consistency, either the methods being proposed should be tested across a large volume of images or a mathematical model must be used to prove their stability. Theoretically, the volume of images needed to

prove the consistency of a method is infinite. So, the second option of using a mathematical model is used in this research. The demonstration of the key concepts in this thesis is presented mathematically by using the results from the equations presented in this section.

4.4. Constraints of the Problem

This section of the report aims at understanding the very fine details of the research. This section investigates the three major constraints that determine the quality and stability of the metrics to be designed. After an extensive analysis of existing methods the constraints are determined to be: -

- The implications of handling fields that are not spatially equivalent
- The limitation in using various processes in the design flow due to a lack of compatibility with the frequency domain (Fourier Transform)
- The parameters that contribute to the stability and instability in the image domain.

4.4.1. Issue of Spatial In-equality

Since the fields are extracted from different spatial locations in the frame, the operations performed on the fields may not be reliable. For example, subtracting the differential equation of two frames with no motion between them must mathematically yield zero values or null residual (assuming no compression noise), but if this operation happens in the field layer, then the residual will have both low and high frequency components. If the residual frame contains just high frequency components, then its impact could be considered to be negligible, as a simple low pass filter will bring the residual values to virtually zero. However, the presence of low frequency components will contribute to the unreliability of the results.

Using the results presented in section 4.3, the issue of spatial equivalency can be illustrated mathematically. Sub-sampling the frame with even lines (4-3-2) into a field with half the vertical resolution gives the following equation (4-4-1): -

$$F_{EF}\left(w_x, \frac{w_y}{2}\right) = \frac{1}{2}F\left(w_x, \frac{w_y}{2}\right) + \frac{1}{2}F\left(w_x, \frac{w_y}{2} + \Pi\right) \quad (4\text{-}4\text{-}1)$$

Shifting the frame with odd lines (4-3-3) downwards by one line before sub-sampling results in the following equations (4-4-2, 4-4-3, 4-4-4): -

$$F_O\left(w_x, w_y\right) e^{j w_y} = \left(\frac{1}{2}F\left(w_x, w_y\right) - \frac{1}{2}F\left(w_x, w_y + \Pi\right)\right) e^{j w_y} \quad (4\text{-}4\text{-}2)$$

$$= \frac{1}{2}F\left(w_x, w_y\right) e^{j w_y} + \frac{1}{2}F\left(w_x, w_y + \Pi\right) e^{j(w_y + \Pi)} \quad (4\text{-}4\text{-}3)$$

$$= \frac{1}{2}F\left(w_x, w_y\right) e^{j w_y} - \frac{1}{2}F\left(w_x, w_y + \Pi\right) e^{j w_y} \quad (4\text{-}4\text{-}4)$$

Sub-sampling the frame with shifted odd lines into a field with half the vertical resolution gives the following equation (4-3-5): -

$$F_{OF}\left(w_x, \frac{w_y}{2}\right) = \frac{1}{2}F\left(w_x, \frac{w_y}{2}\right) e^{j\frac{w_y}{2}} + \frac{1}{2}F\left(w_x, \frac{w_y}{2} + \Pi\right) e^{j\frac{w_y}{2}} \quad (4\text{-}4\text{-}5)$$

Subtracting the sub-sampled odd (F_{OF}) and even fields (F_{EF}) gives the following equations (4-4-6, 4-4-7): -

$$res = F_{EF} - F_{OF} \quad (4\text{-}4\text{-}6)$$

$$res = \frac{1}{2}F\left(w_x, w_y\right)\left[1 - e^{j\frac{w_y}{2}}\right] + \frac{1}{2}F\left(w_x, \frac{w_y}{2} + \Pi\right)\left[1 - e^{j\frac{w_y}{2}}\right] \quad (4\text{-}4\text{-}7)$$

It can be observed from equation (4-4-7) that the residual resulting from subtracting two fields with no motion between them is not zero and they have distinct low and high frequency components. This issue can be mitigated only by performing the operations in the frame layer, which can be accomplished by interpolating the fields to frame resolution. If the interpolated fields are subtracted from each other, the base band term will disappear, but the alias term will still be

present, which indicates the presence of high frequency noise. This will be explained mathematically in detail in section 4.4.3. This section established the necessity to operate in the frame layer rather than the field layer; the following section looks into the choice of interpolation methods available.

4.4.2. Interpolation Methods and Frequency Domain Compatibility

The process of interpolation is known as de-interlacing. Various interpolation methods are used for de-interlacing; the challenge is to select the best interpolation method for the proposed research scenario. The basic principle behind de-interlacing is that the reconstructed frame must not have any jittery artefacts. Popular linear interpolation methods have already been explained in Chapter 2. Popular non-linear methods are based on edge detection and motion estimation. The principle behind the edge detection methods is to estimate the edge direction of each pixel and change the interpolation direction accordingly (Lee et al., 2007; Lee, 2008, Park et al., 2003). The motion estimation algorithm detects the section of the frame with motion and applies the relevant interpolation algorithm, (Ding et al., 2006; Kim et al. 2002; Nicolas et al. 2008). There are some hybrid algorithms; these apply de-interlacing using some untraditional methods by switching between multiple de-interlacing methods, for example, utilising temporal interpolation for static areas and spatial interpolation for motion areas (Oh et al., 2000; Won et al., 2007; Chen et al., 2004).

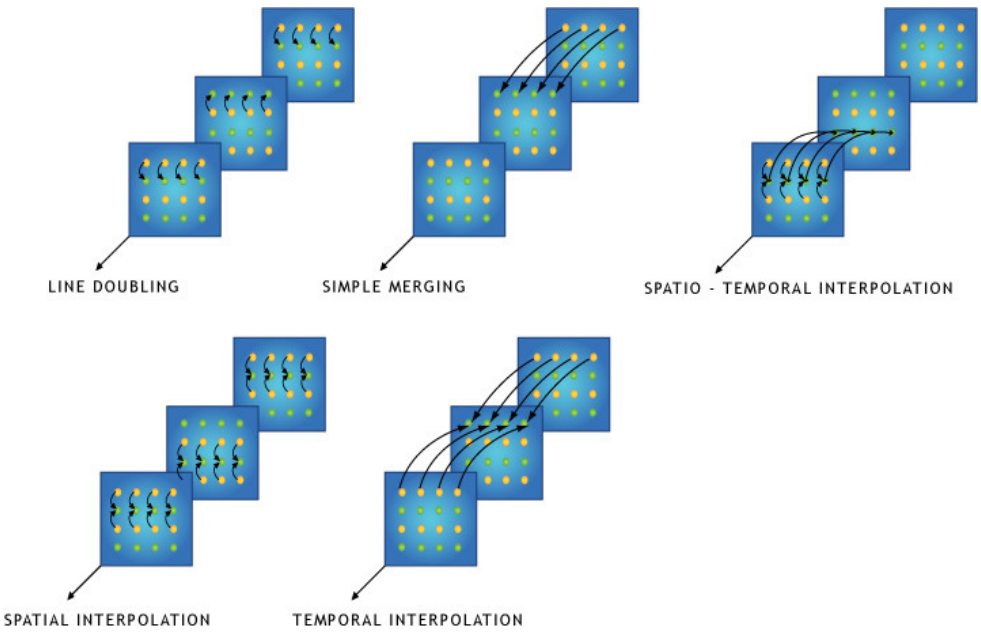


Figure 4-3. Interpolation methods (Mallat, 2006)

The problem with using the mathematical frequency model for the inter-field motion analysis is that the reference model assumes motion to be linear and equal for every pixel in the field. In reality, the motion is not linear unless there is zooming, camera panning or a scene change. This results in a big restriction in the choice of interpolation methods that can be used for the problem. Since the mathematical model assumes there is either linear motion or no motion at all, the interpolation method used must be linear.

This section emphasises the need to use a linear interpolation method for up-converting a field into a frame for mathematical stability. Most of the higher layer algorithms in existence utilise a linear interpolation method of some kind in the system. The method may be spatial, temporal or spatio-temporal depending on the system requirements. Various linear interpolation methods are graphically illustrated in Figure 4-3 and their corresponding frequency spectrums are illustrated in Figure 4-4. A new terminology 'error frame' is introduced, which is the difference between interpolated top and bottom fields. The next section of the thesis investigates parameters that contribute to the stability and instability of the metrics to be designed, if different systems choose to use different interpolation methods to generate an error frame.

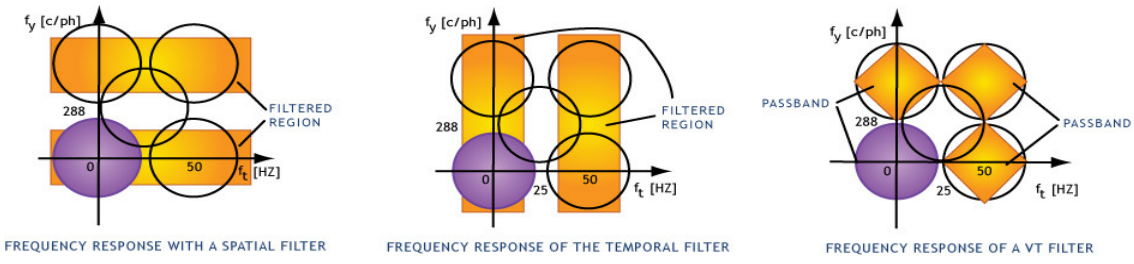


Figure 4-4. Frequency response of de-Interlacing filters (De Haan and Bellers, 1998)

4.4.3. Stability Analysis of the Error Frame

The main aim of this section is to explain the characteristics of the error frame. The end product of the research is expected to be a metric of high consistency across

images of varying characteristics. To design a metric of such high accuracy, it is mandatory to understand and identify the parameters that change and remain static in the error frame regardless of the interpolation operations performed on them. Two interpolation methods, 'spatial' and 'temporal', are translated into the frequency domain for analysis using the results of Beuker and Shah (1994), from section 4.3. The derivations are very lengthy and complicated, so only the important results are presented in this section. The detailed derivation is presented in Appendix B.

The first interpolation method translated into the frequency domain is spatial in nature. Spatial interpolation is a process where the missing lines are patched by averaging the spatial neighbours. Equation (4-4-8) represents the interpolated values of the missing lines of the even line frame.

$$F_{EA}\left(w_x, w_y\right) = \frac{1}{4}F\left(w_x, w_y\right)\left[e^{-j w_y} + e^{j w_y}\right] + \frac{1}{4}F\left(w_x, w_y + \Pi\right)\left[e^{-j w_y} - e^{j w_y}\right] \tag{4-4-8}$$

Equation (4-4-10) represents the spatially interpolated version of the sub-sampled even field.

$$F_{ES}\left(w_x, w_y\right) = F_E\left(w_x, w_y\right) + F_{EA}\left(w_x, w_y\right) \tag{4-4-9}$$

$$F_{ES}\left(w_x, w_y\right) = F\left(w_x, w_y\right)\left[\frac{1}{2} + \frac{e^{-j w_y}}{4} + \frac{e^{j w_y}}{4}\right] + F\left(w_x, w_y + \Pi\right)\left[\frac{1}{2} - \frac{e^{-j w_y}}{4} - \frac{e^{j w_y}}{4}\right] \tag{4-4-10}$$

Equation (4-4-11) represents the interpolated values of the missing lines of the odd line frame.

$$F_{OA}\left(w_x, w_y\right) = \frac{1}{2} F\left(w_x, w_y\right)\left[\frac{e^{-j w_y}}{2} + \frac{e^{j w_y}}{2}\right] + \frac{1}{2} F\left(w_x, w_y + \Pi\right)\left[\frac{e^{-j w_y}}{2} - \frac{e^{j w_y}}{2}\right] \quad (4\text{-}4\text{-}11)$$

Equation (4-4-13) represents the spatially interpolated version of the sub-sampled odd field.

$$F_{OS}\left(w_x, w_y\right) = F_O\left(w_x, w_y\right) + F_{OA}\left(w_x, w_y\right) \quad (4\text{-}4\text{-}12)$$

$$F_{OS}\left(w_x, w_y\right) = \frac{1}{2} F\left(w_x, w_y\right)\left[1 + \frac{e^{-j w_y}}{2} + \frac{e^{j w_y}}{2}\right] - \frac{1}{2} F\left(w_x, w_y + \Pi\right)\left[1 - \frac{e^{-j w_y}}{2} - \frac{e^{j w_y}}{2}\right] \quad (4\text{-}4\text{-}13)$$

The second interpolation method translated into the frequency domain incorporates the temporal coefficients in addition to the spatial coefficients. The spatio-temporal interpolation is a process where the missing lines are patched by averaging both temporal and spatial neighbours. Equation (4-4-14) represents the temporally interpolated values of the missing lines of the even line frame.

$$F_{EST}\left(w_x, w_y\right) = F\left(w_x, w_y\right)\left[\frac{1}{6} + \frac{e^{-j w_y}}{6} + \frac{e^{j w_y}}{6}\right] - F\left(w_x, w_y + \Pi\right)\left[\frac{1}{6} + \frac{e^{-j w_y}}{6} + \frac{e^{j w_y}}{6}\right] \quad (4\text{-}4\text{-}14)$$

By adding the shifted frame with the interpolated values to the original sub-sampled even line frame, the spatio-temporal interpolated even line frame is given by equation (4-4-16).

$$F_{EVT}\left(w_x, w_y\right) = F_E\left(w_x, w_y\right) + F_{EST}\left(w_x, w_y\right) \quad (4\text{-}4\text{-}15)$$

$$F_{EVT}\left(w_x, w_y\right) = F\left(w_x, w_y\right)\left[\frac{2}{3} + \frac{e^{-j w_y}}{6} + \frac{e^{j w_y}}{6}\right] + F\left(w_x, w_y + \Pi\right)\left[\frac{1}{3} - \frac{e^{-j w_y}}{6} - \frac{e^{j w_y}}{6}\right] \quad (4\text{-}4\text{-}16)$$

The process is repeated again for spatio-temporal interpolation on the odd line frame. Equation (4-4-17) represents the temporally interpolated values of the missing lines of the odd line frame.

$$F_{OST}\left(w_x, w_y\right) = F\left(w_x, w_y\right)\left[\frac{1}{6} + \frac{e^{j w_y}}{6} + \frac{e^{-j w_y}}{6}\right] - F\left(w_x, w_y + \Pi\right)\left[\frac{1}{6} + \frac{e^{j w_y}}{6} + \frac{e^{-j w_y}}{6}\right] \quad (4\text{-}4\text{-}17)$$

Adding the shifted frame with the interpolated values to the original sub-sampled odd line frame. The spatio-temporal interpolated odd line frame is given by equation (4-4-19).

$$F_{OVT}\left(w_x, w_y\right) = F_O\left(w_x, w_y\right) + F_{OST}\left(w_x, w_y\right) \quad (4\text{-}4\text{-}18)$$

$$F_{OVT}\left(w_x, w_y\right) = F\left(w_x, w_y\right)\left[\frac{2}{3} + \frac{e^{j w_y}}{6} + \frac{e^{-j w_y}}{6}\right] + F\left(w_x, w_y + \Pi\right)\left[-\frac{1}{3} + \frac{e^{j w_y}}{6} + \frac{e^{-j w_y}}{6}\right] \quad (4\text{-}4\text{-}19)$$

The spatial and spatio-temporal interpolations reveal some interesting results. There is a numeric ratio between the methods (4-4-20, 4-4-21, 4-4-22). This clearly establishes that the phase or distribution of the signal remains constant regardless of

the interpolation method used; the parameter that is subject to variation is the luminance value. It can also be observed from equations (4-4-23, 4-4-24), that subtracting the interpolated odd and even field removes the base band term. This justifies the claim in section 4-4-1 that the low frequency components will disappear if operated in the frame layer.

$$F_{OS}\left(w_x, w_y\right) = \frac{2}{3} F_{OVT}\left(w_x, w_y\right) \tag{4-4-20}$$

$$F_{ES}\left(w_x, w_y\right) = \frac{2}{3} F_{EVT}\left(w_x, w_y\right) \tag{4-4-21}$$

$$F_{ES}\left(w_x, w_y\right) - F_{OS}\left(w_x, w_y\right) = \frac{2}{3}\left[F_{EVT}\left(w_x, w_y\right) - F_{OVT}\left(w_x, w_y\right)\right] \tag{4-4-22}$$

$$F_{OS} - F_{ES} = F\left(w_x, w_y + \Pi\right)\left[1 - \frac{e^{-j w_y}}{2} - \frac{e^{j w_y}}{2}\right] \tag{4-4-23}$$

$$F_{OVT} - F_{OVT} = 2F\left(w_x, w_y + \Pi\right)\left[-\frac{1}{3} + \frac{e^{j w_y}}{6} + \frac{e^{-j w_y}}{6}\right] \tag{4-4-24}$$

It is clear from the discussion that a consistent metric must be independent of the luminance value variations. The distribution of pixel values in the error frame must be quantified rather than the intensity values. The lack of consistency in existing methods is because they are similar to traditional objective metrics, which are luminance value dependent. The following section presents an in-depth analysis of the error frame by investigating the connectivity characteristics of the high and the low frequency components within the image.

4.5. Distribution of Low and High Frequency Components in an Error Frame

This section investigates the high and low frequency components in the error frame. It is very important to know the contribution of each pixel in an error frame. Each pixel can be classified into either low or high frequency components based on their distribution and the connectivity exhibited towards their neighbours. This process plays a key factor in designing the metric, as the pixels accounting for the

high frequency components must make no contribution to the metric calculation as it would result in inconsistency.

Previously, it was explained that an error frame is the difference between interpolated even and odd fields. The mismatch between interpolated fields with no motion is more likely to happen with the diagonal edges than with the horizontal and vertical edges. In other words, the error frame will have high frequency residue if the edge directions take values other than 90^o and 180^o. The inter-field motion translates into low frequency residue in the error frame and occupies the low frequency spectrum. The size of the cluster and distribution of the pixel values reflect the velocity and the direction of the motion respectively. A pixel can be classified as a low or high frequency component by analysing its connectivity characteristics. A cluster of pixels with high connectivity can be considered to be an object (Jan and Lin, 2002).

After the extensive analysis of the error frame characteristic, the high frequency components are defined as follows: “A pixel can be considered a high frequency component if it belongs to a cluster whose size is less than a 4 x 4 non-zero pixel matrix”. Similarly, “A pixel can be considered a low frequency component if it belongs to a cluster whose size is greater than or equal to a 4 x 4 non-zero pixel matrix”. From the discussion, it is evident that the smallest identifiable low frequency component is a 4 x 4 pixel matrix. This heuristic size is determined after an examination of the error frames of many images containing objects of varying curvature at very close proximity to the screen. In reality, the size of the smallest identifiable object is variable, as the spatial frequency of the object varies with the dimension of the screen and the distance of the observer from the screen. This issue will be dealt with in later sections; at the moment, the algorithm will assume a low pass cluster of fixed size.

The error frame of an object with no inter-field motion will have discontinuous pixels along the object edges. The error frame of an object with inter-field motion will have a solid mass of pixels exhibiting a high level of connectivity; in other words, they will be close clustered. There is a high level of similarity in the structure of the error frame generated by using traditional edge operators and the method used

in the proposed research. The core difference lies in the proposed filter's ability to handle highly compressed video frames, as they are low pass filtered and exhibit a certain degree of blurring. The ramp between the gray level transitions will be broader in compressed images. Applying simple gradient operators will produce broader edges traversing into the low frequency spectrum of the error frame. The proposed method is capable of producing significant boundaries onset and offset of the ramp if the image is blurred. The proposed method is analogous to a second derivative Laplacian operator. The principle difference between the gradient and the Laplacian operator is that the gradient gives finite values all through the ramp and the Laplacian operator gives values at the onset and offset of the ramp. The broader ramps increase the distance between the onset and offset of the Laplacian values while limiting the residual impact within the high frequency spectrum. Figure 4-5 compares the behaviour of different image operations in generating an error frame, while operating on compressed images.

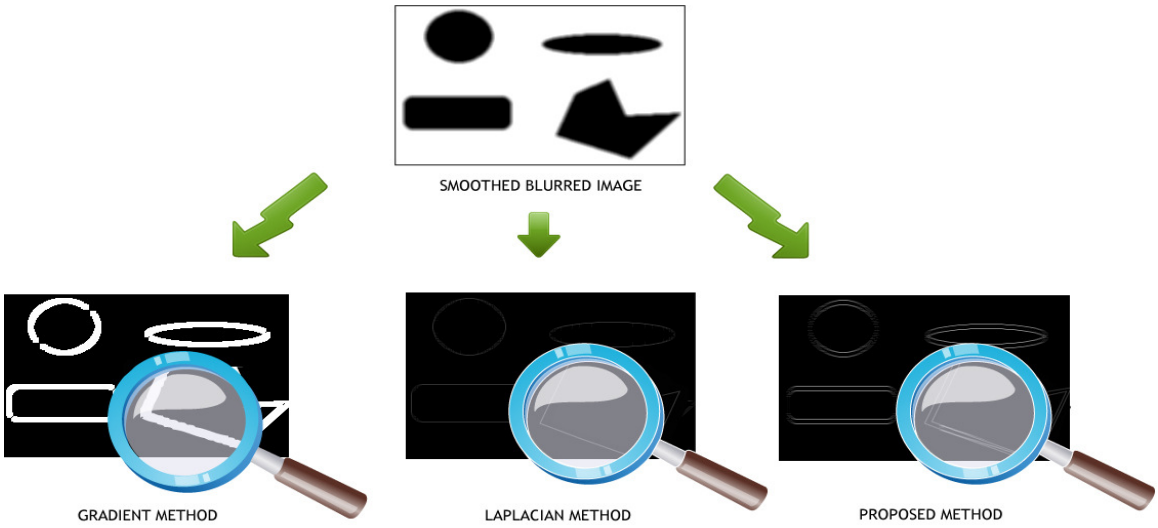


Figure 4-5. Comparison of different methods on a compressed blurred image

The similarity between the Laplacian operator and the proposed method applies only to compressed images, as their responses with interlaced frames with inter-field motion are completely different. When applied to objects with inter-field motion, the response of gradient and Laplacian methods interchange as the edges are one pixel in length in the vertical direction and the Laplacian method will have broader edges

than the gradient method. In the proposed method, the low frequency component of the error frame results in an irregular shape depending upon the direction and speed of the moving object. The irregular shaped object in the error frame will be filled with luminance values resulting from finding the average of the values from the top and bottom fields. This will be a near uniform region with minimal variance. This is consistent with the findings of Mustafa and Sethi (2005), specifically, that moving objects leave a fading trail behind them, and the gradient indicates the direction and the length of the trail indicates the speed. Figure 4-6 compares the behaviour of different image operations in generating an error frame while operating on frames with inter-field motion.

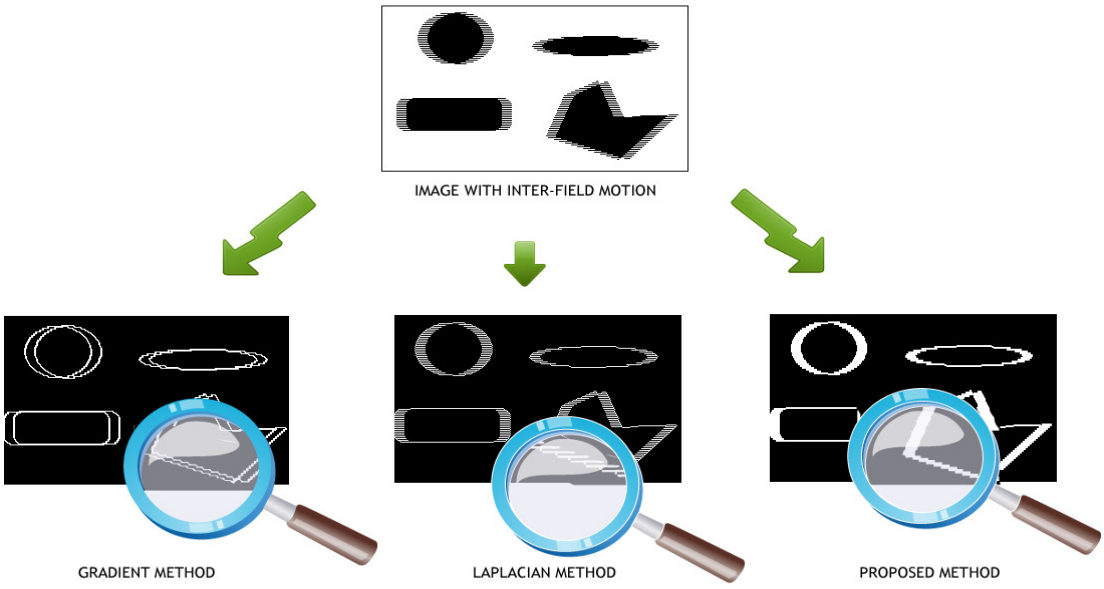
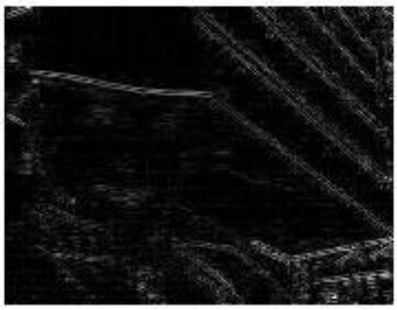


Figure 4-6. Comparison of different methods on an image with inter-field motion

Noise is an important factor that needs special attention when operating on compressed images. Since the frames are compressed with different quantisation parameters, the reconstruction error or compression noise will show up in the error frame. Apart from the compression error, the interpolation error will also lead to noise in the error frame. The video compression is guided by the rate matching algorithm (balance between bitrate and quality) during transcoding. Since the research deals with a commercial problem, the videos to be tested are usually of good quality, as the broadcasters are required to follow certain quality standards. Even if

the editing factories ignore the need to set the quality standards while encoding, the rate matching algorithms have an in-built module to control the quality of the compressed video. The rate matching algorithms normally clip the QP (Quantisation Parameter) value between 0 and 31 and ensure that the QP of the frame currently being coded does not vary from the previous frame's QP by more than 25% (ISO 14496-2, 2002). This process ensures consistency in the compression process when dealing with variable bitrate video sequences. From experimentation, it has been found that a choice of scaling factor between 8 and 10 removes the compression noise from the error frame.

No inter field motion



High frequency components

Inter field motion

Low and high frequency components

Figure 4-7. Low and high frequency components

Since the structure of the objects with and without inter-field motion is known, the next step is to quantify them within a consistent range. Revisiting the concepts from section 4.5, the intensity or luminance values must not be considered for the metric design process; it is the distribution and the connectivity of the pixels in the error frame that hold the vital information. Figure 4-7 shows the high and low frequency components of a frame with and without motion. In the following sections,

some novel methods for quantifying the distribution of pixel values are proposed that are in line with above the theories.

4.6. Novel Metrics for Quantifying Inter-Field Motion

The primary challenge is to extract the object of interest from the pool of low and high frequency components and represent them in a meaningful way. Much research has been carried out in the field of object extraction and representation. The methodology of some general algorithms in existence includes the usage of the Hough transform (Mustafa and Sethi, 2005), variable frame windows (Hwang et al., 2004), pre-stored background information (Alsaqre and Baozong, 2003), edge delineation (Jan and Lin, 2002), the Gaussian mixture model (Zhang and Lu, 2001) and multi-layer processing (Liu and Chen, 2008).

The object extraction problem was different in the research. Due to the real time nature of the problem, the time constraints restricted the usage of filters and other image processing operations that demand high image storage and additional processing time. This restricted the usage of morphological transforms and advanced representation methods. The core objective was to neutralise the high frequency components present in the error frame and quantify the low frequency components in a meaningful way. The pre and post processing modules could not be included in the design due to a lack of hardware resources.

Novel mathematical operations are explained to solve the above-mentioned issues. The critical aspect of designing a mathematical model is to be consistent across different image resolutions, qualities and standards. The mathematical model of the error frame applies theories on elliptical and circular structures to the field of image processing. An ellipse is defined as a locus of points in a plane, where the sum of distances to two fixed points is a constant. A circle is defined as a plane, in which the distance from any point on the plane to the centre is a constant. An ellipse becomes a circle when the major axis equals the minor axis. The circle-ellipse pair is similar to the square-rectangle pair.

Any irregular shaped object can be approximated into a rectangle or an ellipse. After experimentation with irregular shapes, it was found that approximating an irregular object to an elliptical model will yield more precise results than approximation to a rectangular model. The 'circularity' metric is investigated in depth, as the metrics proposed for quantifying inter-field motion will be in line with this metric. The circularity of an object defines the circular characteristics of an object (how circular an object is), and has been used in different forms (Scotney et al., 2001). Ireton and Xydeas (1991) used circularity to avoid the under classification and over classification of an object. The formula used by Ireton and Xydeas (1991) for measuring circularity is given by equation (4-6-1).

$$Circularity = \frac{Area}{(Perimeter)^2} \tag{4-6-1}$$

Kilday et al. (1993) used circularity to represent the characteristics of Mamographic lesions. The formula used by Kilday et al. (1993) for measuring circularity is given in equation (4-6-2).

$$Circularity = \frac{(Perimeter)^2}{Area} \tag{4-6-2}$$

Di Ruberto and Dempster (2000) used a complicated combination of methods for accurate ROI (Region of Interest) processing. The formula used by Di Ruberto and Dempster (2000) for measuring circularity is given in equation (4-6-3).

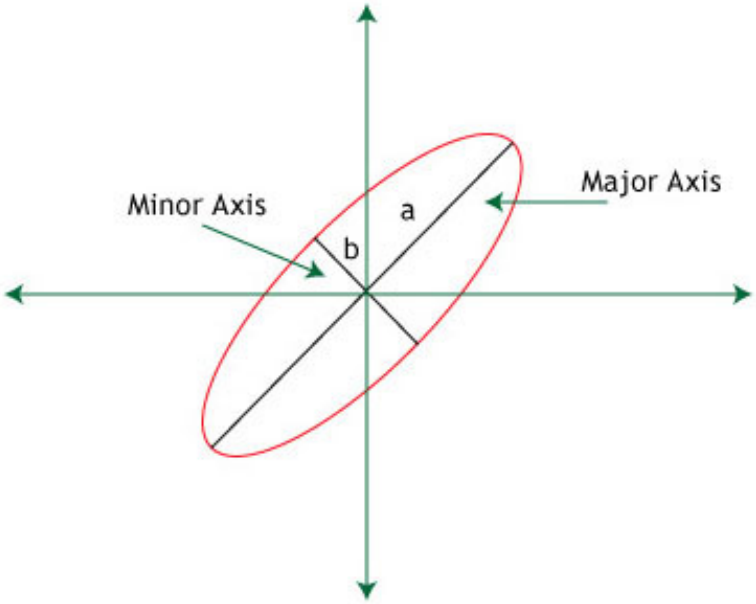
$$Circularity = 4\pi \frac{Area}{(Perimeter)^2} \tag{4-6-3}$$

This section illustrates some very interesting properties of an ellipse. Most of the operations in the mathematics can be performed in the image domain by using special image operators. The area and the circumference of a circle are defined by equations (4-6-4, 4-6-5) respectively, where 'r' is the radius of the circle.

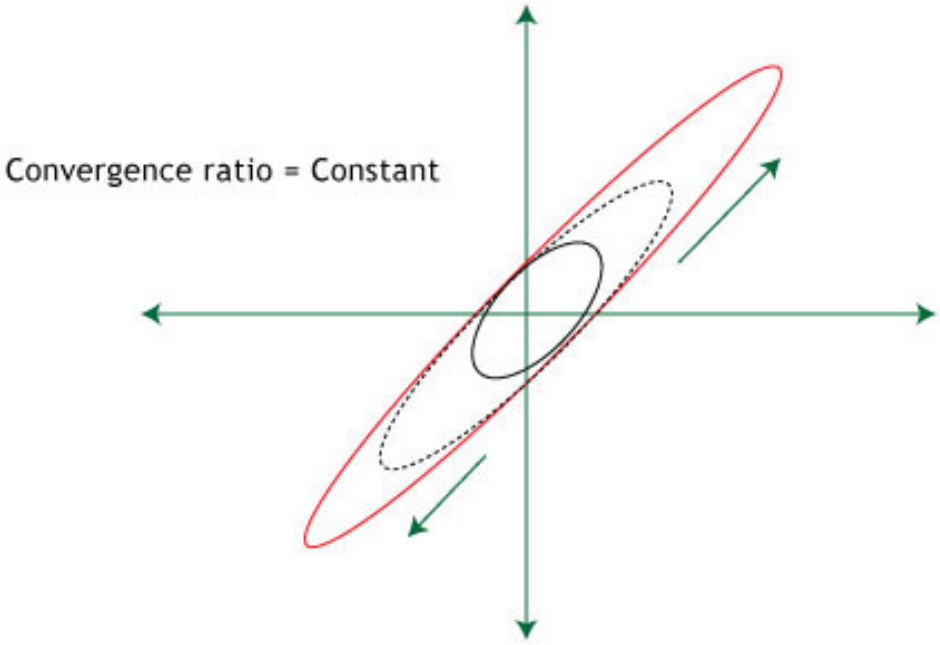
$$Area_{circle} = \pi \ r^2 \qquad (4\text{-}6\text{-}4)$$

$$Circumference_{circle} = 2\pi \ r \qquad (4\text{-}6\text{-}5)$$

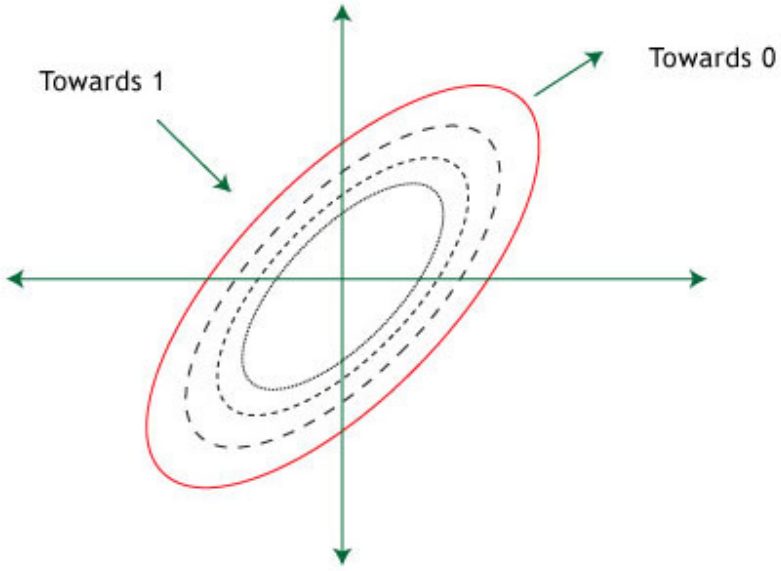
The area and the circumference of the ellipse are defined by equations (4-6-6, 4-6-7), where 'a' is the major axis and 'b' is the minor axis. The structure and the parameters of the ellipse are illustrated in Figure 4-8.a.



(a)



(b)



(c)

Figure 4-8. Properties of an ellipse

$$Area_{ellipse} = \pi \; a \; b \tag{4-6-6}$$

$$Circumfere\ nce_{ellipse} = 2\,\pi \sqrt{\frac{a^2 + b^2}{2}} \tag{4-6-7}$$

The area of the ellipse equals the circumference, while the major axis of the ellipse equals the minor axis with their values being 2. In other words, the condition of convergence occurs when the ellipse shrinks to a circle with a radius equal to 2 (4-6-8, 4-6-9).

$$\begin{cases} Area_{ellipse} < Circumference_{ellipse} & when \;\; a,b < 2 \\ Area_{ellipse} = Circumference_{ellipse} & when \;\; a,b = 2 \\ Area_{ellipse} > Circumference_{ellipse} & when \;\; a,b > 2 \end{cases} \tag{4-6-8}$$

$$Area_{ellipse} = Circumference_{ellipse}$$

$$\pi \; a \; b = \; 2 \, \pi \; \sqrt{\frac{a^2 + b^2}{2}}$$

$$2\,a^2 + 2\,b^2 - a^2\,b^2 = \; 0$$

$$a = b = r = 2 \qquad \text{(4-6-9)}$$

From the above results, a novel relation 'convergence ratio' (4-6-10) is proposed. The convergence ratio is the ratio between the circumference and the area of an ellipse. The convergence ratio decrements towards zero when the area of the ellipse increases (4-7-11). The convergence ratio increments when the area of the ellipse decreases and reaches a maximum value of 2 (4-6-11). This is illustrated in Figure 4-11.c. This point of convergence has been used to solve the research problem, as the circumference is less than the area before the convergence point and the circumference is greater than the area after the convergence point.

$$Convergence\ ratio_{ellipse} = \frac{Circumference_{ellipse}}{Area_{ellipse}} = \frac{2\sqrt{\frac{a^2 + b^2}{2}}}{a\,b} \qquad \text{(4-6-10)}$$

$$Convergence\ ratio_{ellipse} = \begin{cases} \geq 1 & when \;\; a,b \leq \; 2 \\ < 1 & when \;\; a,b \; > \; 2 \end{cases} \qquad \text{(4-6-11)}$$

The eccentricity of the ellipse is dependent on the major and the minor axes. If the minor axis of the ellipse is held constant and the major axis is increased, the convergence ratio remains the same. Similarly, if the major axis of the ellipse is held constant and the minor axis is increased, the convergence ratio remains the same. This is illustrated in Figure 4-8.b. The convergence ratio and its scale invariant

property are the core principles behind the metrics proposed in the following section for inter-field quantification.

4.6.1. Convergence Ratio Metric

In an error frame, a cluster of pixel values with high connectivity can be considered to be an object. The area of the object can be estimated by the sum of all the pixel values (4-6-14). The circumference can be calculated by applying the gradient operator (4-6-13) to the image and summing the absolute values. The ratio between the gradient and the sum will give the convergence ratio (4-6-12).

$$Convergence\ ratio_{image} = \frac{Gradient}{Summation} \tag{4-6-12}$$

$$Gradient = \frac{\left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right|}{2} \tag{4-6-13}$$

$$Summation = \sum_{x=0}^{rows}\sum_{y=0}^{cols} F(x, y) \tag{4-6-14}$$

The structure of the high frequency components in the error frame was discussed in section 4.5. The high frequency components occur in clusters smaller than 4x4 non-zero pixel matrices. The pixels constituting the high frequency components can be modelled as an ellipse with a minor axis value between 2 and 3 and an infinite major axis. Infinite major axis implies that the volume of high frequency components in an error frame cannot be predicted due to varying image characteristics. It was discussed earlier that the convergence ratio will remain constant if either of the axes is held constant and the other is incrementing. So the error frame populated by high frequency components will have a constant convergence ratio and will remain within the range regardless of the volume of the high frequency components.

The convergence ratio of an ellipse decreases with any increase in the major and minor axes. So, if the convergence ratio of the image is smaller than that of an ellipse with a minor axis of 3 and an infinite major axis, then this indicates the

presence of a bigger cluster, which could have been due only to inter-field motion. In the presence of inter-field motion, the convergence ratio will move towards zero as the inter-field motion increases.

Another important issue that needs addressing is that the above mathematical model assumes that the objects are of constant intensity, but in reality they are not. The objects are of different texture distributions and so are the values in the error frame. In mathematical terms, the gradient operator is analogous to differentiation and the mean operator is analogous to integration (4-6-15). If the convergence ratio is calculated for an object with a constant texture and an object with varying texture, the convergence ratio will still be approximately the same if the object areas were constant. This can be justified by the fundamental theorem of calculus:

"Integration and differentiation are inverse operations".

$$Convergence\ ratio_{image} = \frac{Gradient}{Summation} = \frac{\left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right|}{2\sum_{x=0}^{rows}\sum_{y=0}^{cols} F(x,y)} \qquad (4\text{-}6\text{-}15)$$

Any change in the texture values affects both the numerator and the denominator in the convergence ratio. This relation keeps the convergence ratio from fluctuating, so the convergence ratio of an object of specific dimensions will be approximately the same regardless of the texture variations within the image.

If spatial interpolation is used, then the convergence ratio of a frame with only high frequency components or no inter-field motion will range between 0.45 and 0.90; any value smaller than this indicates the presence of inter-field motion. If the spatial interpolation is replaced by spatio-temporal interpolation, the range and the cut-off will still be the same. This is because of the fundamental theorem of calculus, as any change in the method used for generating an error frame will have exactly the same impact on both the numerator and the denominator. This shows the flexibility of the metric, as it is not necessary to use a specific interpolation method; the threshold value will remain a constant regardless of the interpolation method being used, as long as it remains a linear operator.

The output of the quantifier using the convergence ratio could be modelled using an inverted Gaussian or Normal distribution with the threshold cut-off being the mean. Since it is difficult to specify a definite value for the threshold across different resolutions, a transition range is specified, where frames could be either progressive or interlaced with minimal artefacts, which justifies the normal distribution curve illustrated in Figure 4-9.

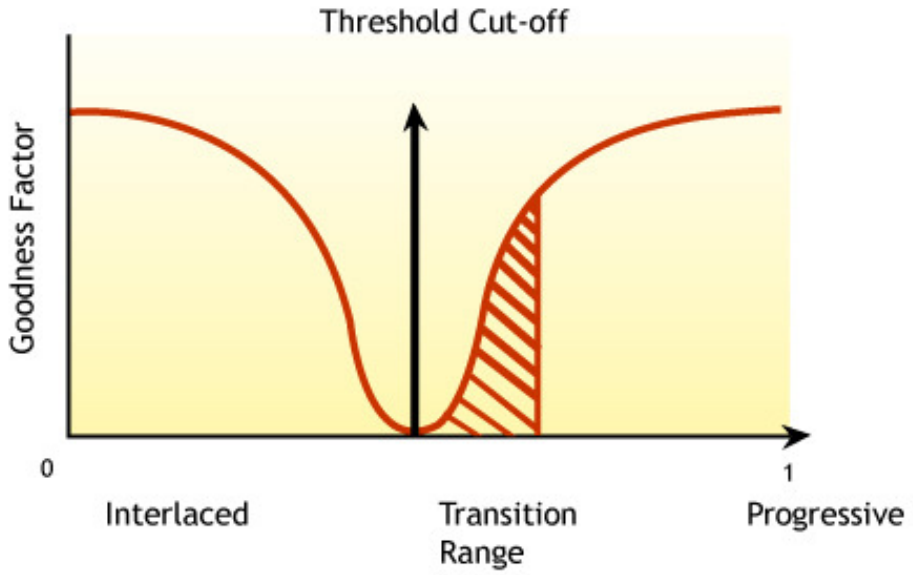


Figure 4-9. Inverted Gaussian curve for convergence ratio

The convergence ratio is a consistent metric and will produce the same results across different resolutions. However, when human eyes view images at different resolutions, the spatial frequency varies. The spatial frequency represents the complexity of the image. The spatial frequency measured in cycles/degree is a function of the viewing angle and the distance of the observer from the screen. When the display resolution increases, the viewing angle increases as well; this in turn increases the pixel density represented per degree viewing angle. So this implies that when a convergence ratio is applied to two images of different resolutions and the value generated is the same, then the area occupied by the inter-field motion is the same in both images. Interestingly, the impact of the inter-field motion may not perceptually be the same to human eyes due to the difference in resolution. As a consequence, the convergence ratio may be interpreted differently based on the screen resolution and the average viewing distance of the observer.

This section has reviewed the convergence ratio metric for quantifying inter-field motion accurately. The convergence ratio quantifies the inter-field motion in a frame regardless of the influence on the observer; in other words, it is a physical metric. For the effectiveness of the higher layer algorithm, it might be just sufficient to process the frames that have a visual impact on the observer. This section also explained the method of interpretation of the convergence ratio on images of varying resolution. The next section of the thesis explains another novel metric in which advanced human perceptual principles are integrated in the design process of the metric.

4.6.2. Gradient Deviation Ratio Metric

In real-time systems, where higher layer algorithms are used to improve the perceptual quality of the video by processing the frames on the fly, the processing speed can be greatly improved by selecting the appropriate frames to process. To accomplish this, a novel metric 'Gradient Deviation Ratio' is proposed, which is a modification of the convergence ratio by incorporating some principles of human visual perception, so that only frames that have a significant impact on the observer's perception are classified as interlaced.

Figure 4-10. Low spatial frequency frame (0.3847)

In Figures 4-10 and 4-11, both images have approximately the same amount of inter-field motion, which is also reflected in the convergence ratio values shown in the braces, but the inter-field motion is masked by the image complexity in the second image. This implies that the two images have the same amount of inter-field motion, but differ in the level of perceptual impact they have on the observer. The human perception of a video is dependent on two different factors: 'spatial masking' and 'temporal masking'. Masking happens when the true influence of a signal on a human observer is suppressed by the presence of another signal. When masking of a pixel value happens due to the pixel's neighbours in the same frame, then the

phenomenon is known as ‘spatial masking’; subsequently, when the masking of pixel values happens due to the temporal neighbours in the neighbouring frames, then the phenomenon is known as temporal masking. The issue with application of the spatial and masking functions to the research problem is that the problem is neither spatial nor temporal. An interlaced frame viewed on a progressive screen is a spatial problem, whereas the same frame viewed on an interlaced display is a temporal problem.

Figure 4-11. High spatial frequency frame (0.3820)

The primary factors affecting the spatial masking of a pixel value are average background luminance and spatial non-uniformity of the background luminance (Chou and Chen, 1996; Netravali and Prasada, 1977; Yang et al., 2005). The primary factor affecting the temporal masking is the inter-frame difference (Chou and Chen, 1996; Yang et al., 2005; Lee and Dickinson, 1994). The temporal masking influence

will not be considered in the metric design process. Since the temporal masking is used to predict the influence of a pixel on the observer over a series of frames (if the inter-frame difference is low, then the influence of the pixel on the observer is low and vice versa), it would be necessary to buffer multiple frames for processing, which is not suitable for real-time processing.

Spatial masking occurs when either the gradient of the pixel background is large or the ratio between the background luminance and the pixel value is smaller than the Weber ratio (visibility threshold). The methods of spatial and temporal masking explain the masking characteristics of an individual pixel with respect to the frame parameters. In the research problem, it is the combined masking effect of the spatial frequency on the inter-field motion in a frame that should be addressed rather than the pixel-by-pixel influence. On analysis of the images of different spatial frequencies and inter-field motion, it could be established that the following argument agreed by researchers on human perception holds true- *The image activity of higher spatial frequencies are less visible than that of lower spatial frequencies* (Chou and Chen, 1996).

The gradient deviation ratio metric takes the spatial detail of the image into account and quantifies the inter-field motion accordingly. This is accomplished by replacing the summation in the denominator of the convergence ratio with the mean absolute deviation of the error frame. For an image with a constant background texture or very low spatial detail, the mean absolute deviation will be approximately the same as the summation. This is because the background luminance values in the error frame will be negligible when compared to the luminance values resulting due to inter-field motion. Whereas, for an image with complex background or high spatial detail, the mean absolute deviation will be lower than the simple summation. This is because the background luminance values in the error frame will be as big as the luminance values resulting due to inter-field motion. This will place the metric in the scale appropriately, reflecting the perceptual impact on the human eye (the higher the spatial detail, the lower the mean absolute deviation and vice versa). For a constant inter-field motion in a frame, if background complexity is increased, the convergence ratio will remain constant, whereas the gradient deviation ratio will move towards the cut-off, which reflects the masking of inter-field motion due to

increased spatial frequency. The mean absolute deviation will vary based on the resolution as the mean value is influenced by the dimension of the image. The gradient deviation ratio must be interpreted differently based on resolution, but the threshold will remain constant across images of same resolution.

The following equations (4-6-16, 4-6-17) illustrate the gradient deviation ratio calculation, where μ is the mean value.

$$Gradient\ Deviation\ Ratio_{image} = \frac{Gradient}{Mean\ Absolute\ Deviation} \tag{4-6-16}$$

$$Mean\ Absolute\ Deviation = \sum_{x=0}^{rows} \sum_{y=0}^{cols} \left| F(x,y) - \mu \right| \tag{4-6-17}$$

Though the convergence ratio and gradient deviation ratio will have a constant cut-off and range regardless of the resolution of the image, these methods have a limitation. The image cannot be broken down into blocks smaller than the image resolution. Since an accurate circumference of the object is necessary for the stability of the ratio, segmenting the image into blocks will distribute the object boundaries across different blocks. This means that these metrics are not suitable to function as a pre-processor for the higher layer algorithms whose methodology is based on block-based processing rather than frame-based processing. To solve this problem, a block-based metric is proposed in the following section in line with the theories utilised for frame-based metrics.

4.6.3. Cluster Ratio

In this section, a block-based metric, ‘cluster ratio’ is proposed where the same principle of measuring the pixel distribution is accomplished, but with an adaptive threshold. The cluster metric (4-6-18) is measured by estimating the number of pixel locations in the error frame that have non-zero neighbours (top, bottom, left, right, diagonals). The notion behind cluster metric is like other artefacts, inter-field artefacts follow the clustering principle, if the combing artefacts occur in clusters (closely distributed), the impact will be distinct, if they are widely distributed, then the impact will be minimum.

$$Cluster\_count = Error\_Frame\begin{pmatrix}(i-1,j);(i+1,j);(i,j+1);(i,j-1);\\(i+1,j+1);(i-1,j-1);(i-1,j+1);(i+1,j-1)\end{pmatrix} > 0 \quad (4\text{-}6\text{-}18)$$

$$threshold = 1.05 * block\_size^{(.6)} \quad (4\text{-}6\text{-}19)$$

It can be observed that the cluster ratio varies with respect to the image area, so an adaptive threshold must be set. The adaptive threshold is computed by using the formula given in equation (4-6-19), which is a heuristic value used in the real-time implementation of the metric. The value was set by manually analysing many blocks with and without inter-field motion. The formula for the threshold cut-off is just an approximation and intense subjective experimentation must be carried out to identify precise threshold estimation and so achieve accurate results. The variation in the convergence ratio with the image resolution was explained in section 4.6.1. However, with the cluster ratio, this issue can be resolved by choosing a variable block size based on the image resolution for perceptual consistency. The cluster ratio is not as stable as the convergence and gradient deviation ratios, but with a reliable higher layer algorithm, it can significantly reduce the computation time of the end-to-end system.

4.7. Summary

Progressive/Interlaced frame classification plays a significant role in many higher layer applications, such as inverse telecine, interlaced to progressive conversion and field dominance detection algorithms. Methods for quantifying inter-field motion/combing artefacts in an interlaced video are proposed in this chapter. The chapter presented mathematical explanations for the lack of robustness in the existing metrics and frequency domain analysis to identify the parameters that contribute to the stability of the filter. Three novel metrics have been proposed, namely, ‘convergence ratio’, which has a constant dynamic range and threshold cut-off across images of different characteristics; ‘gradient deviation ratio’, which incorporates human perception principles to quantify inter-field motion; and ‘cluster

ratio', which exploits the pixel connectivity measures for metric design and provides the flexibility of block-based processing. This chapter has highlighted a unique way of designing an algorithm that is independent of luminance values, in contrast to the current methods, which use pixel differences. The operating limitation revealed at this stage of the research is that the frame should not have any visual blocking or blurring artefacts. The real potential of the metric will be revealed while the field reversal and mixed pulldown algorithms are explained in the next chapter.

5. Field Reversal and Mixed Pulldown Detection

When a higher layer computer vision algorithm is designed to improve the perceptual quality of the video in real-time, the time constraints on offline and online processing differ. The robustness of a real-time video system depends on the ability of the system to process the video frames on-the-fly with the same precision and speed and as that of offline processing. An algorithm that is flawless in principle may not be suitable for real-time implementation due to false positives, lack of accuracy and hardware constraints. In this chapter, a fundamental principle of correlation for detecting field reversal and mixed pulldown frames in a video sequence is gradually modified step-by-step for real-time implementation with high precision. The metrics designed in the previous chapter for quantifying inter-field motion are integrated with the overall system for improving the computation speed of the algorithm.

5.1. Introduction

This chapter explains the methods for removing field reversal and mixed pulldown pattern issues from a compressed bitstream. The previous chapter dealt with deriving a global metric for quantifying inter-field motion. The reason for such intense effort to quantify the inter-field motion precisely rather than using simple metrics will be evident while the field reversal and mixed pulldown detection algorithms are being implemented. Since the algorithms are designed to be implemented in a real time system, emphasis is given to the reduction in the computational complexity and false positives. Since the impact of errors and increased processing time will have a drastic impact on the performance of the commercial product, the methods are designed with the aim of achieving high accuracy. The core principle behind the algorithms is based on image correlation. The unique aspect of the designed system is that both mixed pulldown and field reversal algorithms are integrated into the same system. Intense testing was carried out with Tektronix on a large volume of video test clips, and subsequently, real-time tests with MTV were run before the commercial launch of the test equipment.

5.2. Review of Literature, Problem Definition and Chapter Organisation

There is no literature that directly addresses the problem of field reversal except one US patent (Baylon and McKoen, 2006; Baylon and McKoen, 2008) and a conference paper (Baylon, 2007). The method proposed utilizes a six tap zipper filter (linear shift invariant vertical gradient filter) to classify a frame as being either interlaced or progressive. Further, the first motion estimation is carried out between the top field (even lines) of the current frame and the bottom field (odd lines) of the next frame. Similarly, the second motion estimation is carried out between the bottom field of the current frame and the top field of the next frame. If the first motion value is smaller than the second motion value, then the field order is assumed to be bottom field first. If the second motion value is smaller than the first motion value, then the field order is assumed to be top field first. Baylon and McKoen (2008) also suggested that the same principle may be implemented by replacing the motion estimation module with a shift invariant vertical zipper filter. This is accomplished by combining the bottom field of the current frame and the top field of the next frame into a new frame; similarly, the top field of the current frame and the bottom field of the next frame are combined to form another new frame. If the volume of the zipper points in the new frame 'one' is more than that in new frame 'two', then the field order of the two frames is considered to be top field first; if not, then the field order of the two frames is considered to be bottom field first.

Though the simulation graphs showed good performance statistics, the method has some serious drawbacks. The method utilises multiple motion estimation, which is costly in terms of implementation. Further, it is a very general method with many heuristic assumptions; the method assumes the video sequence as interlaced and does not address issues pertaining to hybrid video sequences (coexistence of different video types). The method has been designed for offline processing rather than online processing. Figure 5-1 shows the graphical illustration of the method proposed by Baylon and McKoen (2006).

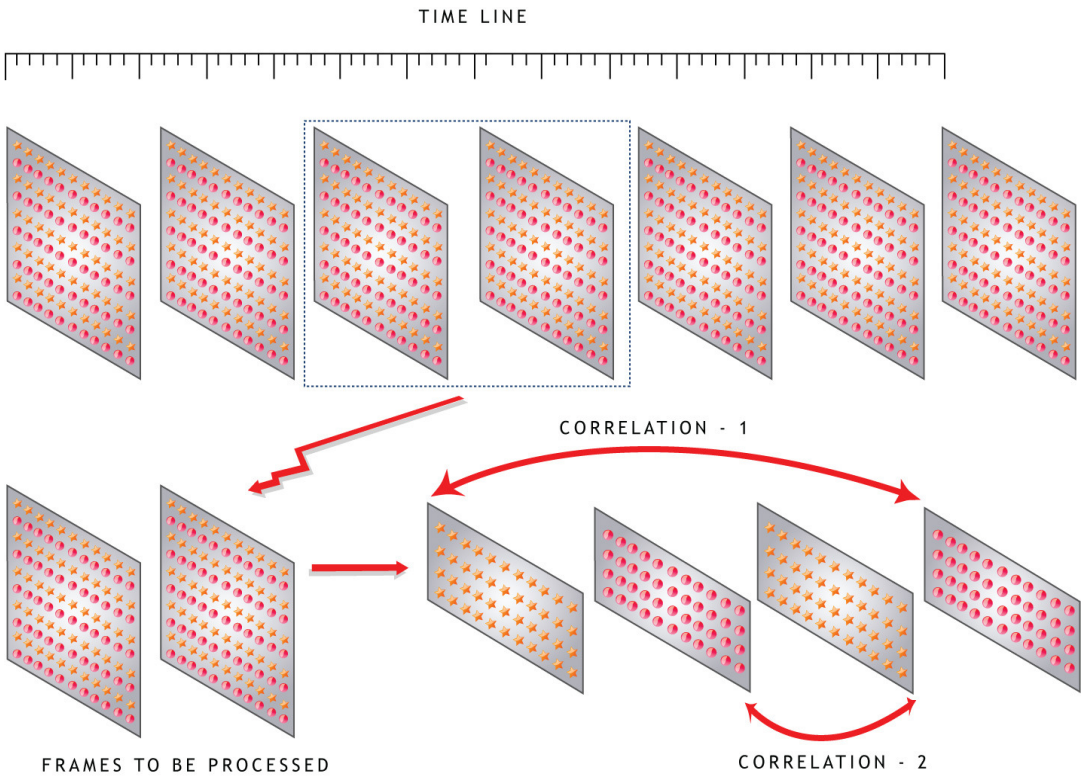


Figure 5-1. Method by Baylon and McKoen (2006)

There is an extensive amount of literature addressing the problem of pulldown detection. All the methods are based on the fundamental principle of calculating the difference between successive fields. The methods could be broadly classified based on the timeline, as in the methods proposed in the late 1990s, early 2000s and after. The methods proposed in the early 1990s accumulate and analyse the field difference values every five fields; if one of the values shows a large deviation from the others then the corresponding field is assumed to be redundant. These methods were proposed in the analogue television era, before progressive televisions and digital broadcasting had made an impact. Gove et al. (1995) implemented the traditional method with a threshold decision circuit for easy hardware implementation. Casavant et al. (1994) used a median filter to avoid large spikes due to scene change and a correlation filter to reduce false positives due to noise. Coombs et al. (1996) improved the robustness of the traditional method by incorporating an adaptive threshold. Wells (1998) explained a method by which the correlation method can be implemented with minimum storage. Other methods identified redundant fields using slightly different methods. Correa and Schweer (1994) estimated whether the amplitudes of the pixel values are monotonically increasing or decreasing over

successive fields, a method that was later used to capture the redundant fields. Christopher and Correa (1997) made the decision on a pixel-by-pixel basis by incorporating both vertical pixels of the adjacent field to detect a pulldown frame. Keating and Richards (1995) used a motion estimation method to calculate the correlation between the fields. The video stream must be identified as a pulldown stream before the algorithm is applied. If an ordinary progressive stream that is fed into the system having been erroneously identified as a pulldown stream, the algorithm will still process the stream, but will stamp the frames with minimal motion as pulldown frames. Another major issue is the application of the algorithm to hybrid video sequences, in which the interlaced, progressive and pulldown frames coexist. The existing algorithms will throw out false positives, as they cannot cope with the type changes within a stream unless pre visual/manual identification occurs.

The methods proposed from the early 2000s on are designed to handle hybrid videos, as progressive videos had penetrated the market and the hybrid videos had started to dominate. Conklin (2006) dropped the frames with inter-field motion, and the time elapsed between successive drops was used to decide if the video sequence had been pulled down. Hui (2005) partitioned the frame into zones and the parity of the difference values in the zones was used to determine if the frame was interlaced or pulldown. Matsubara and It (2008) used absolute field difference values between three successive fields to identify a pulldown frame. In all the methods, many thresholds are used in the decision-making process; these thresholds are known as ‘magic numbers’ in the broadcasting industry. This leads to a fundamental problem of spatial and temporal image changes influencing the precision of the algorithm. The threshold value is set to a heuristic value or a safe margin, which leads to fluctuation in the performance of the algorithms (the methods perform extremely well with a specific sequence and completely fail with another sequence).

The previous literature review highlights that some heuristic assumptions are used in the development of the algorithms. The resulting video from the pulldown process will have a uniform mixture of interlaced and progressive frames. Though the principles behind the derivation of interlaced and pulldown frames differ, they are identical in structure. From the previous discussion, it is understood that every interlaced frame must have a field order, but the exception is with the pulldown

frames. The pulldown frames are not actual physical frames; they are virtual frames created by displaying a physical field for longer than its display time. Since pulldown frames are not actual frames, they cannot have a field order. For example, in the MPEG-2 standard, when the *Repeat_First_Field* is set, then the *Top_Field_First* flag will not convey the field order information; instead, it will convey the repetition rate of the field (if set to '0', the field is repeated twice and if set to '1', the field is repeated thrice).

Revisiting the mixed pulldown issue explained in Chapter 3, when multiple pulldown patterns coexist in a video stream, the application of a traditional reverse telecine algorithm will remove the wrong frames. This will result in a video with a down-sampled frame rate from 29.97 frames/sec to 24 frames/sec, but the pulldown frames will still be present in the stream. When this erroneous video is edited or up-sampled again, it will result in a video stream that lacks integrity. When reverse telecine methods are used for processing the video stream, searches are made for a fixed number of pulldown patterns once in every five fields. However, when these methods are applied to the mixed pulldown stream, complications arise as false positives are generated. When the existing field reversal algorithm is applied to the hybrid video frames, it tags the pulldown frame with a field order, and as a result, the source of the frame is lost for ever.

Once the field order is assigned to a frame, it is considered as interlaced thereafter. Consequently, it is imperative that the field order algorithm and the mixed pulldown detection algorithms be integrated into a single module. It is an added benefit that the process in hybrid videos is automated. The inter-field quantifier proposed in Chapter 4 will classify the frame as being either progressive or interlaced, and subsequently, the field reversal/mixed pulldown detection algorithms will categorise the pre-classified frame as either a pulldown frame or as an interlaced frame with the correct field order.

This section illustrates the primary requirements of a robust real-time system. The efficiency of the real-time system depends on the speed and the memory utilization. The operations that dynamically allocate memory of a variable size will slow down processing and have a drastic influence on performance. This restricts the

number of video frames that can be stored during processing. The algorithms that are designed to be implemented in a real-time system must be capable of processing the frames 'on-the-fly' and be totally independent of any other frames in the video sequence. Existing algorithms suggest methods such as processing random frames; processing the first few frames; and processing only frames with large motion, to conclude with the field order and pulldown pattern of the video sequence. The major reasons why these approaches are not appropriate are listed below: -

- The existing methods may be suitable when the video sequence is processed off-line, but in real-time, it is not possible to move back and forth along the sequence and conclude with appropriate results.
- Not all the terminal devices can run the algorithm and change the flags in the bitstream. The primary job of the terminal equipment like DVD players is to read the stream and display it accordingly; they may be able to change the method of display (interpolation method, field order and so on), but cannot modify the source video file.
- If the results are concluded after processing a few frames sequentially or randomly, the methods will fail with hybrid videos. This is because it is impossible to predict the nature of the stream automatically without decoding the whole stream; if the algorithm processes the first few frames, then it makes a wrong assumption that the video stream is not hybrid in nature.

For a method to be highly flexible, it must process the frames on-the-fly and produce the best possible results, with no bias or presupposition. The conditions for the best possible results are listed below: -

- Every interlaced frame with inter-field motion, however small, should be assigned the right field order.
- Every pulldown frame in the video stream must be signalled by flagging the *Repeat_Field* flag.
- A 24 frames/sec stream with mixed pulldown errors must be treated as an all progressive sequence.
- The algorithm must be capable of processing the video sequence at twice the frame rate for real-time performance.

The chapter starts by revisiting the concepts from Chapter 4 on issues regarding the operation on field and frame layers. Section 5.4 explains the influence of random motion in the video sequence on the precision of the algorithm, and subsequently investigates the optimal frame window size to mitigate the problem. Section 5.5 explains the core design flow of the field reversal and mixed pulldown detection algorithms with an in-depth analysis of each module. Section 5.6 explains the moving average window technique, the aim of which is to reduce the number of false positives. Section 5.7 explains the optimisation procedures carried out to adapt the proposed methods to suit real-time implementation. The integration of the inter-field quantifier proposed in Chapter 4 is also explained in this section and section 5.8 concludes the chapter with a summary.

5.3. Field domain vs Frame domain

Though the principle of operation of the algorithm proposed by Baylon and McKoen (2006) is theoretically able to solve the problem, in practice, the technique generates many false positives and thus is not suitable for commercial implementation. In the previous chapter, the drawbacks of existing methods for interlaced/progressive classification were rectified by designing a robust inter-field quantifier. The reasons for the generation of false positives in the method proposed by Baylon and McKoen are critically analysed and modifications are made to improve the precision of the algorithm. Rigorous testing of the proposed algorithm with videos of different standards, resolutions and qualities is explained.

The method proposed by Baylon and McKoen (2006) and the methods for the pulldown detection operate in the field layer. It has already been explained in Chapter 4 that it is necessary to interpolate the frames prior to any operations. The same principle applies to field reversal/mixed pulldown methods as well. Since the adjacent fields contain lines from different sample lattices, the field difference will result in an error (Li et al., 2000).

An abstract detail of the mathematical proof explained in Chapter 4 is given below: -

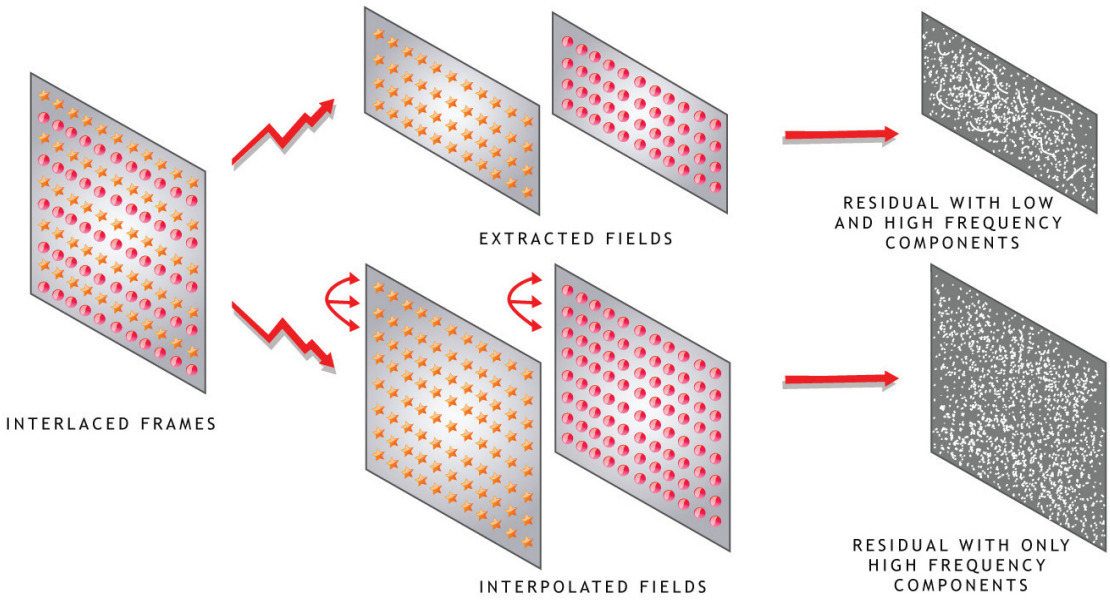


Figure 5-2. Frame and field residual

$$res_{field} = F_{EF} - F_{OF} \tag{5-3-1}$$

$$res_{field} = \frac{1}{2} F\left(w_x, w_y\right)\left[1 - e^{j\frac{w_y}{2}}\right] + \frac{1}{2} F\left(w_x, \frac{w_y}{2} + \Pi\right)\left[1 - e^{j\frac{w_y}{2}}\right] \tag{5-3-2}$$

$$res_{frame} = F_{ES} - F_{OS} \tag{5-3-3}$$

$$res_{frame} = F\left(w_x, w_y + \Pi\right)\left[1 - \frac{e^{-j w_y}}{2} - \frac{e^{j w_y}}{2}\right] \tag{5-3-4}$$

It can be seen from equations (5-3-2) and (5-3-4) that subtracting the fields with no inter-field motion results in both high and low frequency residue, whereas subtracting frames (interpolated fields) results only in high frequency residue. Figure 5-2 shows the graphical illustration of these issues.

5.4. False Positives due to Linear Motion Assumption

This section illustrates how the occurrence of random motion in the frames can challenge the precision of the algorithm. Explanation is also given of the need to increase the minimum frame window size to make a reasonable decision to avoid false positives. In the field order detection method proposed by Baylon and McKoen (2006), the correlation is applied to two frames to determine the field order. From the explanation given in section 5.2, it can be established that the method gives reliable results only if the motion in the frames is linear. If the frames exhibit random non-linear motion, the results produced by the algorithm are reversed and the results in false positives.

The primary reasons for this problem are that the field order decision is made simultaneously for two successive frames and there is no way of determining the nature of the motion in the frames (linear or non-linear). This also generates many false positives, which in turn challenges the success of the algorithm.

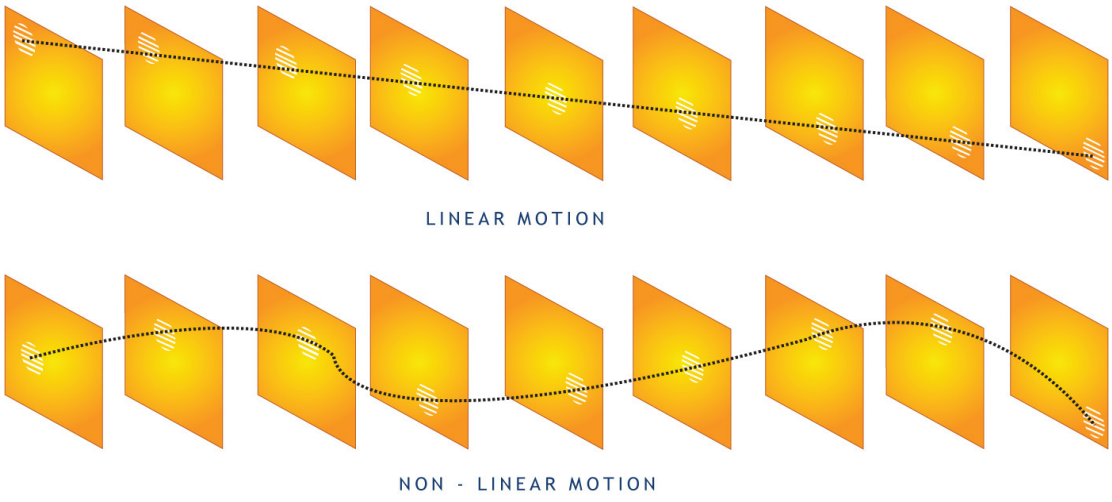


Figure 5-3. Linear and non-linear motion

The solution to the above problem is to process one frame at a time and increase the size of the frame window (Lee et al., 2007). A frame window expanded from 2 to 3 is found to be suitable. This results in two correlation domains (current frame-previous frame and current frame-future frame). If the motion is linear, then

the results produced by the two domains must be the same; if not, then the motion is considered to be non-linear and the results are not reliable, so further processing is required to evaluate the field order. Figure 4-3 shows the graphical representation of the linear and non-linear motion issue.

5.5. The Core Principle and Implementation

The solution to the problem is based on the spatial and temporal correlation between the successive frames of the video. 'Correlation' could be defined as the degree of similarity between two frames. This principle was implemented by Baylon and McKoen (2006), where the authors measured the correlation between the fields in successive frames using the motion estimation principle and the zipper filter. The correlation between adjacent frames is likely to be higher compared with the correlation with frames separated by a distance. The same theory could be applied to the field-based video, where each frame constitutes two fields that belong to different time instants. Applying the above principle to our problem, the field to be displayed first would have a closer relation to the preceding frame and the field to be displayed second would have a closer correlation to the succeeding frame. The importance of operating in the frame domain rather than in the field domain to remove low frequency noise was explained in section 5.3. The drawbacks of using a frame window size of 2 and the need to use a frame window size of 3 to avoid the issue of non-linear motion was explained in section 5.4. Incorporating the results from the discussions in the previous sections, the top and bottom fields are interpolated into top and bottom frames and the correlation domain is broken into two. The results of the correlation with past and future frames are required to match for the field order to be determined without an error.

The choice of the correlation metric is very important, as it directly influences the quality of the algorithm. The motion estimation used by Baylon and McKoen (2006) has many drawbacks. For example, if the motion estimation is calculated on a block-by-block basis, the number of block drifts can be given by equation (5-5-1), where 'P' is the search window size.

$$Number_{blockdrifts} = (P \times 2 + 1)^2 \quad (5\text{-}5\text{-}1)$$

At each iteration, the number of subtractions and additions is given by equation (5-5-2), where 'X' is the block size.

$$Number_{adds\ \&\ subs} = (X^2 \times 2) \quad (5\text{-}5\text{-}2)$$

If a field is partitioned into multiple blocks, then the total number of iterations is given by equation (5-5-3), where 'M' and 'N' are the number of rows and columns of the field.

$$Number_{blocks} = \frac{(M \times N)}{X^2} \quad (5\text{-}5\text{-}3)$$

The total number of operations involved in performing the motion estimation between two fields is given by equation (5-5-4).

$$Total\ \ operations = \left[(P \times 2 + 1)^2 + (X^2 + 2)\right]\ \ \frac{(M \times N)}{X^2} \quad (5\text{-}5\text{-}4)$$

It is observed from equation (5-5-4), that the level of complexity involved in performing a correlation using the motion estimation method is intense, and as a result, it is too costly for an economical hardware implementation. Investigations were carried out to find an alternative correlation metric to replace the motion estimation metric without influencing the quality of the results. Careful analysis of the results revealed that simple pixel-by-pixel objective metrics (PSNR, MSE and so on) outperform the motion estimation. The reduction in the computational complexity was approximately in the range of 100:1. Though the there was no change in the number of false positives generated, the pixel-by-pixel metrics were able to process those fields that had very little motion. The frames that yielded inconclusive results with motion estimation gave meaningful results with the objective metrics. This is in line with the argument presented by Li et al. (2000). The reason for there being no reduction in the false positives was the compression noise, as infinite PSNR cannot be achieved in reality with lossy compression, whereas zero

motion vectors are practically possible when there is no significant motion. The number of operations involved in performing a pixel-by-pixel objective correlation is given in equation (5-5-5).

$$Total \quad operations_{\,pixel-by-pixel} = (M \times N \times 2) \tag{5-5-5}$$

The choice of pixel-by-pixel metrics over motion estimation has been widely adopted by other researchers. The importance of detecting the point-wise motion of the objects is emphasized by Kim et al. (2002). Mallat (2006) also disagrees with the fact that the motion estimation techniques are the future of up-conversion, as a single motion cannot be associated with the pixel because object and background motions may be different.

The following summarises the correlation operation. The top and bottom fields are extracted and interpolated into frame resolution by simple spatial methods F_e and F_o are frames with extracted odd and even lines respectively (5-5-6, 5-5-7),.

$$X_T(i,j)_{\substack{i=1,3,5,\ldots m \\ j=0,1,2,3\ldots n}} = \frac{F_e(i-1,j)+F_e(i+1,j)}{2} \tag{5-5-6}$$

$$X_B(i,j)_{\substack{i=0,2,4,\ldots m \\ j=0,1,2,3,\ldots n}} = \frac{F_o(i-1,j)+F_o(i+1,j)}{2} \tag{5-5-7}$$

$$peak = 2^{bits} - 1 \tag{5-5-8}$$

$$PSNR = 10\log_{10}\left(\frac{peak^{\,2}}{MSE}\right) = 20\log_{10}\left(\frac{peak}{\sqrt{MSE}}\right) \tag{5-5-9}$$

The correlation is calculated by the pixel-by-pixel metric (PSNR is used)(5-5-8, 5-5-9) between the previous frame (X_p), future frame (X_f), interpolated top field (X_T) and interpolated bottom field (X_B) (5-5-10, 5-5-11, 5-5-12, 5-5-13). MSE (Mean Square Error) is the square of the difference between two images.

$$Cor\ 1 = PSNR\,(X_p, X_T) \tag{5-5-10}$$

$$Cor\ 2 = PSNR\,(X_f, X_B) \tag{5-5-11}$$

$$Cor\ 3 = PSNR\,(X_f, X_T) \tag{5-5-12}$$

$$Cor\ 4 = PSNR\left(X_p, X_B\right) \quad (5\text{-}5\text{-}13)$$

The previous and the future frames have not been disintegrated into fields, but rather have been treated as whole frames without any interpolation. This is because, if the previous and future frames are to be disintegrated into fields and interpolated, then the field order must be assumed and one of the fields must be selected for interpolation. If there is a manual error in assuming the field order, then this will influence the field order decision of other frames. Secondly, since the primary purpose is to calculate the correlation, the frame made from the combination of the fields gives a strong motion footprint, as it represents the activity that occurred in two time instants. Figure 4-4 shows the graphical representation of the proposed field reversal/mixed pulldown detection method.

In addition, there is a threshold check in place to verify whether the current frame has a significant proportion of data change from the past and the future frames, which is the main principle on which the algorithm operates. This threshold value is denoted by the factors Δ_1 Δ_2; a lower threshold corresponds to the processing of frames with a very minimal correlation and might lead to false positives. By setting an optimal minimum threshold, it is possible to avoid the generation of false positives by objective metrics due to compression noise. A higher threshold corresponds to processing frames with substantial correlation and would reduce the number of false positives, but would also restrict the number of frames to be checked. The minimum threshold limit is calculated by applying the objective metric to static frames and the maximum threshold limit is calculated by applying the objective metric to frames with scene change information. It is important to set the maximum threshold, as this will avoid the processing of frames in which there are fields from two different video sequences; this normally occurs when a scene is cut. Apart from the basic advantage of avoiding false positives, the presence of a pulldown frame is signalled.

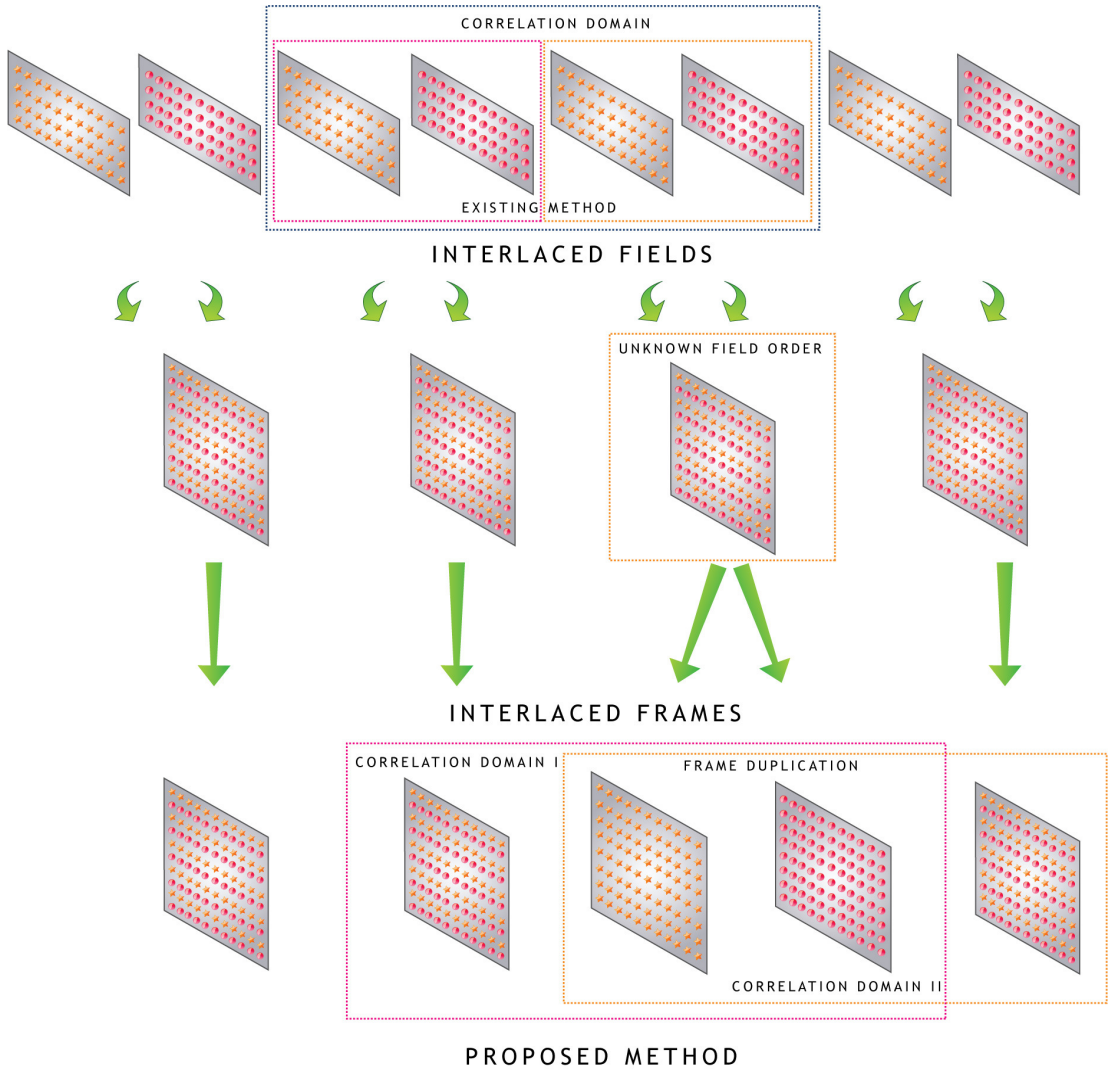


Figure 5-4. Proposed field reversal and mixed pulldown method

$$abs\left(cor\,1 - cor\,3 \right) = \Delta_1 \tag{5-5-14}$$

$$abs\left(cor\,2 - cor\,4 \right) = \Delta_2 \tag{5-5-15}$$

$$threshold\_\min < \Delta_1\ \Delta_2 < threshold\_\max \tag{5-5-16}$$

$$\Delta_1 >> \Delta_2 \tag{5-5-17}$$

$$\Delta_1 << \Delta_2 \tag{5-5-18}$$

While processing a normal interlaced frame, the thresholds Δ_1 and Δ_2 will not differ by a large percentage, but while processing a pulldown frame the thresholds Δ_1 and Δ_2 will differ by a great percentage (5-5-17, 5-5-18) (a difference of 2 dB is the heuristic value set in real time implementation). During the processing of a pulldown

frame, one of the neighbours will be progressive in nature and the other will be interlaced in nature. The interlacing in the neighbouring frame will have been caused by a redundant field and this will cause excessive correlation in one end compared to the other, unless there is a scene change; this has already been dealt by setting the maximum threshold. This situation does not occur with progressive and interlaced frames. The presence of a pulldown frame can be confirmed by applying field-based correlation if necessary. The conditions and inference of the threshold check is shown in Table 5-1.

Table 5-1. Conditions and inference of threshold check

Condition	Inference
$\Delta_1\Delta_2 < threshold\_\min$	No motion
$\Delta_1\Delta_2 > threshold\_\max$	Scene change
$\Delta_1 >> \Delta_2$	Pulldown frame
$\Delta_1 << \Delta_2$	Pulldown frame

Once a pulldown frame is detected, the frame can be handled in two different ways based on the frame rate. If the frame rate is 29.97 frames/sec, then the stream is classified as being pulldown and the frame is removed from the bitstream; subsequently, the *Repeat_First_Field* flag in the relevant location is set. If the frame rate is 24 frames/sec, then it is assumed that the bitstream has been erroneously reverse telecined; the original sequence should have been mixed pulldown, which in turn, contributed to the presence of the pulldown frames in the reverse telecined material. The best way to deal with the problem is to replace the pulldown frame with an interpolated field, so that the integrity of the stream is restored with minimal loss in quality. The stream will now be all progressive, which matches the characteristics of a pure telecine sequence.

If the frame rate is up-sampled again or if the sequence is edited, the resulting sequence will not be corrupt. The method of handling a mixed pulldown frame when

detected differs across different editing factories, which is beyond the scope of this thesis. The objective of the methods proposed in this thesis is to identify the erroneous frames in a video stream, but not to investigate the concealment methods, which is a different domain of research.

It can be observed that, in the proposed new method, an independent check is performed on the frame to detect the combing artefacts, and the cause for the combing is determined by correlating the frame of interest with its neighbours. The independent processing increases the robustness of the system with mixed pulldown patterns, as the proposed method is independent of pulldown patterns and their locations. The method is very efficient compared to existing methods in which a uniform pattern is required for the correct detection of the redundant field. In the proposed new method, the decision is made on a single frame basis, whereas existing methods are based on the store and process principle.

Table 5-2. Conditions and inference of correlation check

Condition	Inference
Cor 1 > Cor 3 && Cor 2 > Cor 4	The field order is top field first
Cor 1 < Cor 3 && Cor 2 > Cor 4	Additional processing required
Cor 1 > Cor 3 && Cor 2 < Cor 4	Additional processing required
Cor 1 < Cor 3 && Cor 2 < Cor 4	The field order is bottom field first

If the threshold check does not indicate the presence of a pulldown frame, then the frame must be an interlaced frame, and the correlation outputs can be used to determine the field order. From the results of the correlation check shown in Table 5-2, it can be seen that condition 1 implies that the field order is top field first and condition 4 implies that the field order is bottom field first. Conditions 2 and 3 shown in Table 5-2 are partial results and there is an equal probability of both bottom and top field being displayed first. This condition normally happens when one of the neighbouring frames is static; the other possibility is when there is a significant

texture change among the frames. The frames of the video stream contain fields that belong to different time instants. The objective metrics operate on the principles of the human visual system: the human eye is not sensitive to individual pixel change, but it is sensitive to contrast change in a large area of the frame. The difference between the frames, for which correlation is calculated, is normalised with the frame dimensions in most of the metrics. This implies that the magnitude of the pixel values has a higher impact than their mode of distribution (volume or density). The objective correlation metric of two frames having a uniform low contrast change will be equal to the objective metric of two frames having a very deep contrast change just in a particular region of the frame. This happens due to magnitude mapping.

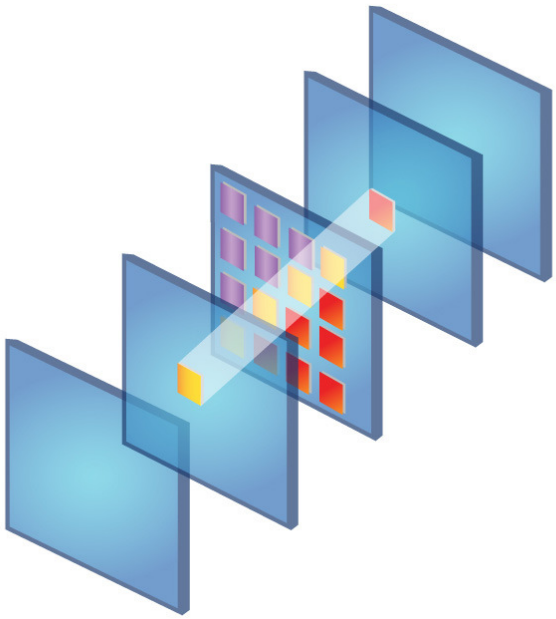


Figure 5-5. Pixel-by-pixel correlation in optical flow metric

$$O = Optical\ \ flow\left(TAR, REF_1, REF_2\right) \tag{5-5-19}$$

$$Opt\ 1 = Optical\ \ flow\left(X_T, X_p, X_f\right) \tag{5-5-20}$$

$$Opt\ \ 2 = Optical\ \ \ flow\left(X_T, X_f, X_p\right) \tag{5-5-21}$$

$$Opt\ \ 3 = Optical\ \ \ flow\left(X_B, X_p, X_f\right) \tag{5-5-22}$$

$$Opt\ \ 4 = Optical\ \ \ flow\left(X_B, X_f, X_p\right) \tag{5-5-23}$$

A novel and simple metric 'Optical Flow Strength (OFS)' to measure the correlation between the frames containing fields from different time instants is proposed. The OFS metric measures the number of pixel points at which the distance of the target frame to reference frame 1 is less than the distance between the target frame to reference frame 2 (5-5-19). The metric ignores the magnitude and estimates just the direction, which solves the above-explained problem of magnitude mapping. Figure 5-5 shows the graphical representation of the optical flow metric calculation. The results of the optical flow check and their corresponding interpretations are presented in Table 5-3.

Table 5-3. Conditions and inference of optical flow check

Condition	Inference
Opt 1 > Opt 2 && Opt 3 < Opt 4	The field order is top field first
Opt 1 < Opt 2 && Opt 3 < Opt 4	Inconclusive result
Opt 1 > Opt 2 && Opt 3 > Opt 4	Inconclusive result
Opt 1 < Opt 2 && Opt 3 > Opt 4	The field order is bottom field first

If the inconclusive result produced by the first correlation method is not rectified by the optical flow method, then the frame's result is stamped as inconclusive. This occurs when there is either no motion or a very random motion between the frames. The results for these frames can be concluded by using a moving average method, which is explained next.

5.6. Moving Average Window for Consistency in the Metadata

The previous sections explained the methods for finding the field order, but there is a possibility that the correlation methods will result in inconclusive results, in spite of there being substantial inter-field motion. To solve this issue, an averaging process, the 'moving windowing' technique, is used to reduce the false positives generated by spurious frames in the stream and inconclusive results. The main

reasons for the occurrence of field order mismatch are usually editing and transcoding mistakes, so it is unlikely for a single frame to exhibit field dominance errors, when previous and successive 'N' frames are error free. For the moving window method, the window size is set to 'K'; then the field order of a frame is the field order that took maximum probability of reoccurrence in the past 'K-1' frames. This process will patch the frames with inconclusive results, as explained in section 5-5.

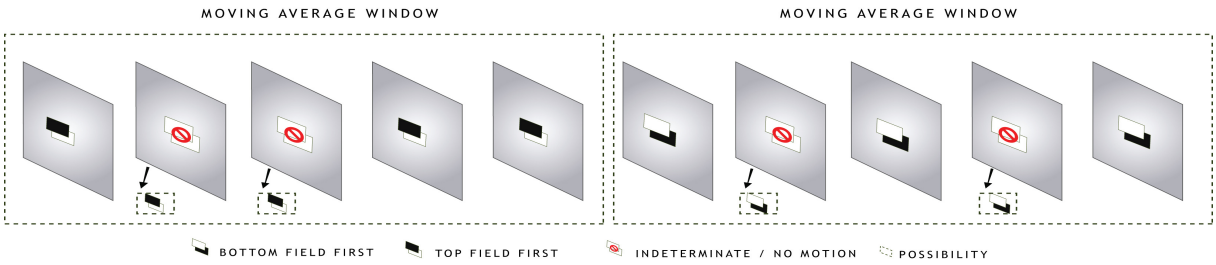


Figure 5-6. Moving Average Window

An additional condition is imposed to avoid field order assignment to the pulldown frames. The moving average window can only be applied, when at least 50% of the frames within the window have a valid field order. This is because when the video stream undergoes the pulldown process, it adds two redundant fields to up-convert four frames into five frames. This will result in two out of every five frames having an interlaced appearance. Even if the field order is assigned to the frames with redundant fields by mistake, the number of frames having a valid field order will not exceed 40% (2 of 5), as 60% (3 of 5) of the frames in the pulldown sequence will be progressive in nature. By choosing the threshold value of 50%, it is possible to avoid field order assignment to frames that, theoretically, must not have a display order. The graphical representation of the moving average window principle is shown in Figure 5-6.

5.7. Performance Optimisation and Pre-processing using Inter-Field Quantifier

The design of the inter-field quantifier in Chapter 4 is now addressed to reduce the complexity of the whole coding process. In the new era of high definition displays, the frame resolution is almost three times the basic standard resolution (720 x 480, 720 x 576). The algorithms that are designed primarily for low resolution images struggle to cope with the increase in the image area and lose their efficiency in terms of speed and computational complexity.

Testing in real-time of an end-to-end prototype indicated that the algorithm was not good enough to be implemented. Many modifications were made to the design to simplify the process. From the previous sections, it can be understood that the field reversal has no impact if there is no inter-field motion between the frames. If the correlation were calculated between the frames with no motion, then it would lead to false positives due to compression noise. These problems can be solved only by using a powerful interlaced/progressive classifier, which can accurately indicate the presence of inter-field motion. This highlights the careful design of the inter-field motion quantifier as described in the previous chapter. The convergence ratio and the gradient deviation ratio provide a powerful filtering mechanism to identify the frames with combing artefacts, which indicates the presence of inter-field motion. Furthermore, the cluster ratio offers the flexibility of partitioning a frame into blocks. The block-based processing will reduce the processing time by processing only those blocks that have vital inter-field motion.

Of the many algorithms in existence that are based on ROI (Region of Interest) processing, the frame algorithm results could be concluded by processing a few blocks with ROI information. Some of the standard applications include block-artefact detection, progressive/interlaced classification, baseband quality checks and computer vision algorithms. A novel method to reduce the number of iterations to increase the speed and reduce power consumption is proposed for those algorithms that use ROI processing as their primary core module. The proposed system does not track any objects; it simply helps to find the first block containing ROI with minimal iterations.

Some of the literature on ROI processing is reviewed in this section. Murching et al. (2005) used a Kalman predictor, which utilised the spiral search method to locate the centroid of the object of interest over a group of frames. The object is selected based on the colour segmentation principle. Trew and Seeling (1994) used a template-based motion tracking, where the object recognition is accomplished by searching a restricted area around its template matching position. Rangan et al. (2001) used an estimation function to estimate the position of the moving entity. All the above-mentioned methods are designed for tracking objects with specific characteristics. The proposed new method is tailor-made for algorithms that utilise block-based processing, which could conclude with a reasonably accurate result after a few blocks with ROI information have been processed. In our scenario, the ROI is the block of the video that contains inter-field motion.

The proposed method is based on four video coding principles: -

- In most cases, the regions of interest between frames are spatially correlated rather than being disjoint
- The majority of the activity happens in the centre of the frame
- Successive frames in a video sequence do not differ by a large percentage
- The displacement of the ROIs between successive frames is minimal.

The proposed method is similar in principle to the three-step search for motion vector estimation used in video compression, where the area of search for the motion vectors is reduced by a random search within the search area; based on the results of the random search, an exhaustive search is performed. In the proposed method, the search area is a frame and the block size is a percentage of the whole frame area. The application of the spiral search method is investigated in the proposed method.

The active region of a frame in a video sequence is identified by performing the spiral search and storing the location of the first ROI block. In succeeding frames, the spiral search method is performed with the previous storage point as the centre.

The size of the spiral window is variable; the maximum iterations are more likely to occur in the first frame, where the first location of ROI is estimated.

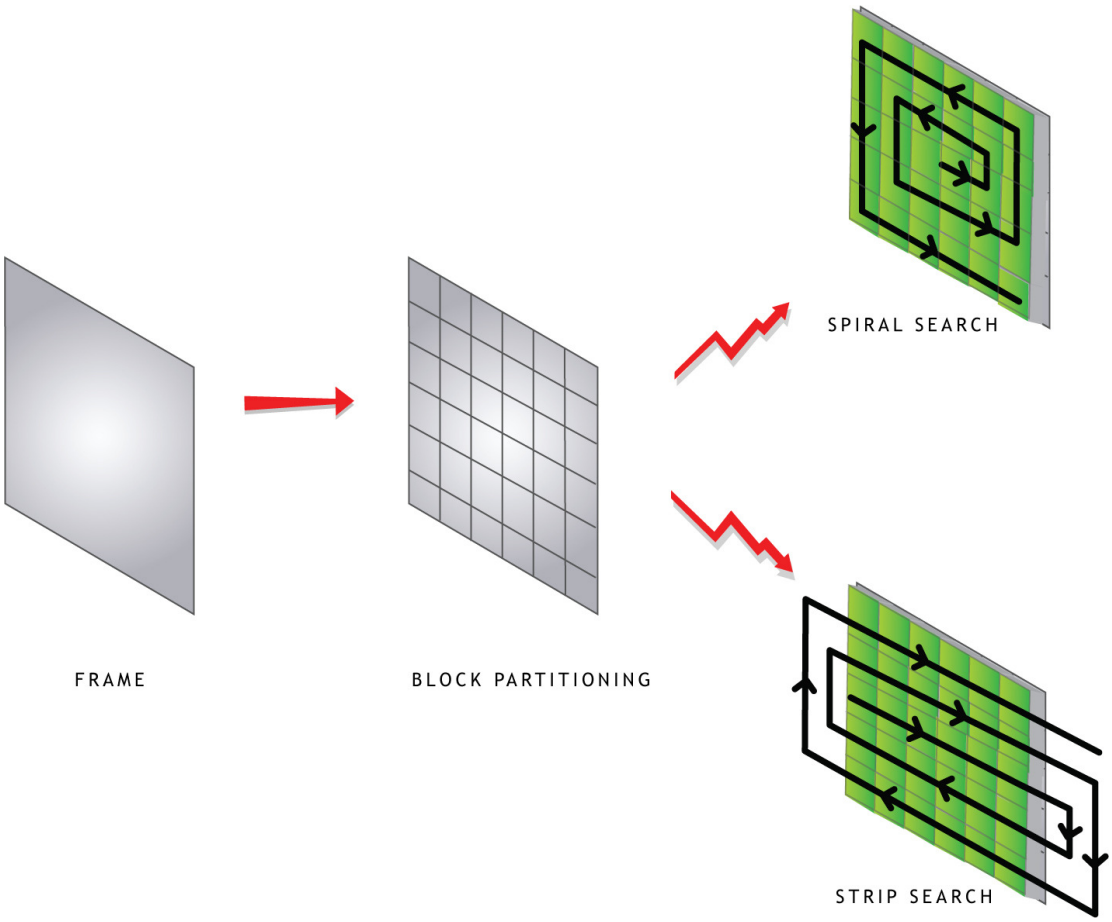


Figure 5-7. Performance optimisation

The determination of the starting direction of the spiral window is decided by the direction of the movement of the spiral window. If the spiral window's direction of movement of the last frame with respect to its previous frame was bottom, then the first block to be processed in the spiral window in the current frame would be the bottom block. The same principle applies if the direction of movement is top, left or right. The spiral search can be replaced by the strip search to reduce the burden on the co-ordinate calculation.

There is no restriction on the size of the blocks that can be used because of the flexibility offered by the inter-field quantifier. The blocks can be picked by either the spiral or the strip search method and, after processing 'n' blocks with reliable motion

information, the decision can be made. The inter-field quantifier scale can be used to estimate the reliability of the blocks by accessing their closeness to the threshold cut-off value. In the commercial version of field reversal detection, a counter is used to facilitate the process; the counter increments as it comes across blocks with the same field polarity and decrements when it comes across blocks with different polarity, hence the result that dominates after the 'n' blocks have been processed is agreed as the final polarity of the frame. Figure 5-7 shows the graphical illustration of the block-based processing schemes using spiral and strip search methods.

5.8. Summary

The objective of the algorithm is to check for the consistency and integrity of the bitstream without manual supervision. The metadata of a compressed bitstream differs with different video compression standards, but most have the concept of field order. In this chapter, 'field reversal' and 'mixed pulldown' algorithms were proposed for solving two major quality problems in the broadcasting industry as a result of metadata inconsistency. The step-by-step approach of modifying a theoretical model to suit practical constraints is presented by incorporating a correlation algorithm, windowing for reducing false positives and spiral and strip search methods for hardware optimisation. The proposed methods analyse the decoded video data and perform image domain operations to detect the field order and rectify mixed pulldown issues, without relying on the metadata in the compressed domain. The results are used to enable automatic quality control to be carried out by performing image analysis and metadata analysis in two domains and checking their consistency. Essentially, the current method of visual inspection in the display domain will be replaced by the proposed system in the source coding domain. The performance of the algorithms proposed in this chapter and the inter-field quantifier proposed in the previous chapter are presented using simulation results in the next chapter.

6. Simulation Results of the Algorithms Proposed to Solve Editing Layer Issues

6.1. Introduction

This chapter explains the simulation results and the real-time testing of the inter-field quantifier, field reversal and mixed pulldown algorithms. Section 6.2 explains the structure of the real-time video testing equipment manufactured by Tektronix, followed by section 6.3, in which the Matlab simulation test bench is explained in detail, along with information on the test clips. Section 6.4 presents the simulation results of the inter-field quantifiers proposed in Chapter 4 and section 6.5 presents the simulation results of the field reversal and mixed pulldown algorithms proposed in Chapter 5. Section 6.6 concludes the chapter with a summary of all the investigations presented in this chapter.

6.2. Structure of 'Cerify'

This section explains the real-time testing platform for the proposed algorithms. 'Cerify' is a Linux-based media testing box for testing the quality of the video before being broadcast. It is a standalone hardware box that is interfaced with the video server where the clips are stored for processing. The Cerify units that are presently in operation are based upon a dual Xeon, rack mounted server made by Intel. Cerify uses the GNU Linux operating system based upon Mandriva Linux and has a 2.6.20 Linux kernel. The user interface is by means of a XML page, where the user can select the clips that have to be processed. When creating a job, the user can select the location of the clip and select the relevant tests to be performed on the clips. For example, a user may choose to detect blocking artefacts, black out frames and freeze frames in a video stream by selecting the relevant check boxes in the user interface. On processing, the system reports the errors in real-time with the relevant frame numbers and information on the exact nature of the error. Multiple clips can be

processed in real-time with the added flexibility of assigning variable priority to different clips.

Figure 6-1. Cerify unit

Figure 6-2. XML user interface

Each test case corresponds to a C++ class; for example, a test case for field reversal is written under the class *FieldReversalTestCase*. The Cerify engine can be started from the command prompt with the location of the video clip and the test case to be run on the video clip. This gives the flexibility of using the Python script to start the Cerify engine from the command prompt with different video clips to run a particular test case. The log files can be written in a text file that could later be used for the performance analysis. Figure 6-1 shows the Cerify unit hardware; Figure 6-2 shows the XML user interface; Figure 6-3 shows the results of the clips on processing; ‘cross mark’ indicates an error, whereas ‘tick mark’ indicates that the

video is error free. Figure 6-4 shows detailed information on the location of the error signalled by Cerify.

Figure 6-3. Processed clips

Figure 6-4. Detailed error information

6.3. Matlab Test Bench

The initial development was undertaken using Matlab and later the programme code was ported into the Cerify system as C++ code, but the real time testing environment did not offer enough flexibility due to following reasons: -

- It was difficult to integrate existing methods that were protected by patents into the system due to IP rights.
- Since the 'test Cerify' machine was shared by other software engineers in the team doing similar testing, it was practically impossible to process one clip at a time, which led to variation in the processing time.
- The algorithms are designed primarily for operating on raw images, but the Cerify test engine is designed for processing compressed video, which results in variable decoding times (H.264 will have a high decoding time, whereas MPEG-2 will have a very low decoding time).

So that a sensible comparison can be made of the simulation data, all the simulations presented in this thesis were run using Matlab. In Matlab, the clips are imported in YUV raw image format and processed with '.m' files. The Matlab software is often very slow at processing compared with 'C++'; the simulations results have a slight deviation from the real time data, but will provide a fair comparison with existing state-of-the-art methods. Figure 6-5 shows the test bench used in the Matlab software, showing the comparison between existing and proposed methods.

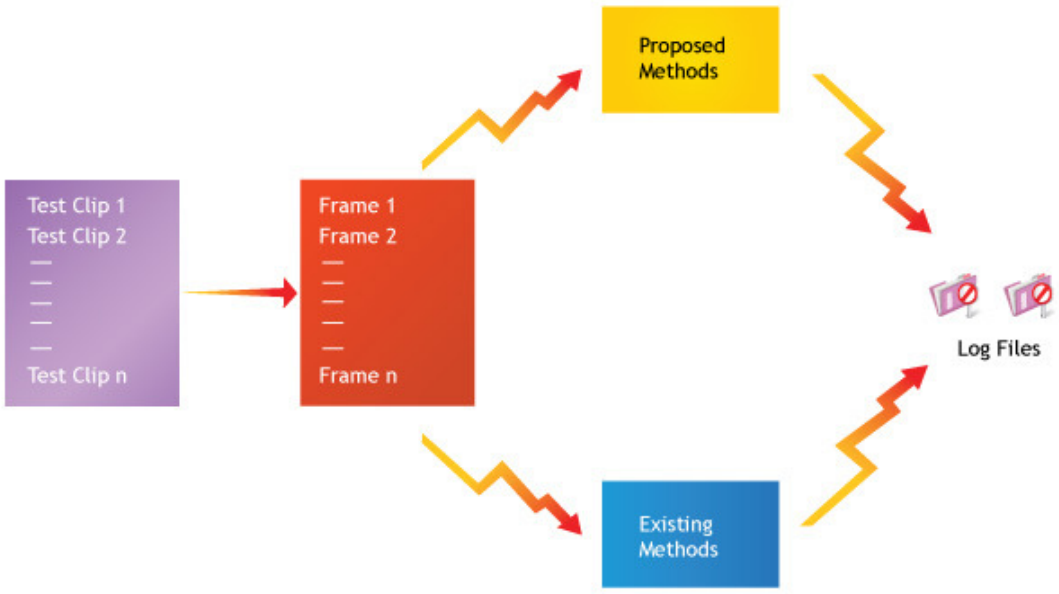


Figure 6-5. MATLAB test bench

About 150 test clips were chosen for testing, as it is very important for the methods to be robust across video clips of different standards, resolutions and qualities. The test clips were chosen from a very broad base, and included both samples from broadcasters and in-house clips. The clips included movies, television material, commercial advertisements, news broadcasts and clips from Tektronix. From the range of clips used for real-time testing, some clips were hand picked and used for simulations in this thesis. The primary reason for selecting the clips was that Matlab cannot handle clips that are of very long duration because of constraints on the processing speed. To avoid these problems, small duration clips with particular varying image characteristics were chosen and decoded into YUV format before processing. Table 6-1 lists all the clips used in the simulations. The characteristics of the clips in terms of resolution, quality, bitrate, compression standard and field order are given in Table 6-1. The video clip is designated using the format *Origin_Name_Type.ext*. 'Origin' refers to the name of the broadcaster or the company from which the clip was acquired, 'Name' signifies the nature of the clip and 'Type' refers to the format of the clip (interlaced, progressive or pulldown). The clips with Origin 'Tek' are the clips picked from the 'Vclips', a collection of clips captured and sold by Tektronix commercially. These clips are captured with a wide variety of spatial and temporal changes and are used by commercial broadcasters to verify the robustness and consistency of the designed algorithms.

The clips listed in Table 6-1 were chosen from a very broad database of picture range and quality. Careful consideration was given to the selection of the test clips, so that they would reflect real world situations with the clips characterised by extreme conditions of spatial, temporal, and quality variations, as shown in Table 6-2.

Table 6-1. List of test clips used for simulations

No	Test clip	Compression Standard	Resolution	Number Of frames
1	ISO_Foreman_Progressive	AVI	176x144	100
2	ISO_Salesman_Progressive	AVI	176x144	100
3	TEK_Person_Progressive	MPEG-2	176x144	168
4	TEK_Women_Progressive	MPEG-2	352x288	50
5	BBC_Video_Progressive	H.264	720x576	98
6	TEK_Flower_Topfield	MPEG-2	704x480	50
7	TEK_Guards_Bottomfield	VC-1	720x576	50
8	CNN_News_Topfield	MPEG-2	720x576	50
9	SKY_News_Topfield	MPEG-2	720x576	50
10	TEK_Lasvegas_Topfield	MPEG-2	320x240	50
11	TEK_1938Movie_Pulldown	MPEG-2	368x480	50
12	MTV_Musicvideo_Pulldown	H.264	720x576	50
13	GOOGLE_Clip3_Pulldown	MPEG-2	528x480	50
14	GOOGLE_Lock_Pulldown	MPEG-2	720x480	50
15	GOOGLE_Moving_Pulldown	MPEG-2	720x480	50

Table 6-2 explains the characteristics of each clip in terms of perceptual quality (good, moderate or poor), type of scene (level of spatial detail), motion information (linear or non linear), object behaviour (foreground and background) and the pre-processing operations involved (cropping, scene cut or interpolation). Most of the algorithms perform optimally with perfectly illuminated high quality images. The important factor for a commercial system is ‘robustness’; the algorithm must produce optimal performance in extreme conditions. To achieve this, rigorous testing was

performed on the proposed methods; every false positive was manually analysed and the issue was resolved.

Table 6-2. Characteristics of the test clips

No	Test clip	Description of the test clip
1	ISO_Foreman_Progressive	Good quality head and shoulder video sequence with some rapid camera movements
2	ISO_Salesman_Progressive	Good quality head and shoulder video sequence with distinct foreground and background
3	TEK_Person_Progressive	Moderate quality video sequence with very fast background movement resulting in blurred appearance of the frame
4	TEK_Women_Progressive	Good quality video sequence with sudden scene change and multiple foreground objects
5	BBC_Video_Progressive	Moderate quality video sequence that has a pixelated appearance due to application of interpolation
6	TEK_Flower_Topfield	Moderate quality video sequence with a very high spatial detail and slow linear uniform motion due to camera panning
7	TEK_Guards_Bottomfield	Good quality and highly interlaced video sequence with distinct foreground and background exhibiting linear motion
8	CNN_News_Topfield	Bad quality and non linear motion resulting in internal interlacing due to texture change within the objects
9	SKY_News_Topfield	Very slight interlacing due to lack of movement of the main object, but visible interlacing in the moving text at the bottom of the scene
10	TEK_Lasvegas_Topfield	Very bad quality video clip with camera panning motion and the video has been cropped from a higher resolution

11	TEK_1938Movie_Pulldown	A 1938 movie telecined from film to digital video and frame rate is up-sampled by pulldown
12	MTV_Musicvideo_Pulldown	Example of a video sequence with mixed pulldown pattern errors; has combination of both interlaced and pulldown frames
13	GOOGLE_Clip3_Pulldown	Moderate quality video stream with mostly pulldown frames and ends with a few interlaced frames
14	GOOGLE_Lock_Pulldown	Moderate quality pulldown video sequence with some scene cuts
15	GOOGLE_Moving_Pulldown	A typical pulldown video sequence with some pulldown frames resembling progressive due to no motion

6.4. Simulation results of Inter-field Quantifiers

Two major metrics were proposed to quantify the inter-field motion: the 'convergence ratio' and 'gradient deviation ratio'; the former physically quantifies the inter-field motion and the latter utilises HVS principles to quantify the inter-field motion in line with human perception. Designing a test bench for testing the robustness of the metrics was very exacting. The clips were disintegrated, which resulted in 850 physical frames. Each frame was visually inspected from different distances and the results were used for testing the proposed metrics.

6.4.1. Performance Comparison of Convergence Ratio, Gradient Deviation Ratio and Cluster Ratio

The convergence ratio, in principle, must capture all the frames in the video sequence with inter-field motion, however small it may be. Each frame was viewed at close proximity to the screen, without any time limitation, and the frames displaying combing artefacts were tagged by storing their frame numbers in a database. Similarly, to test the gradient deviation ratio, the frames were viewed from a reasonable distance from the screen (2 feet) and the frames were displayed on the

screen for less than a second. If the combing artefacts were noticeable then the frame number was stored in the data base. Figures 3-19 and 3-20 show the testing method used for convergence ratio and gradient deviation ratio respectively.

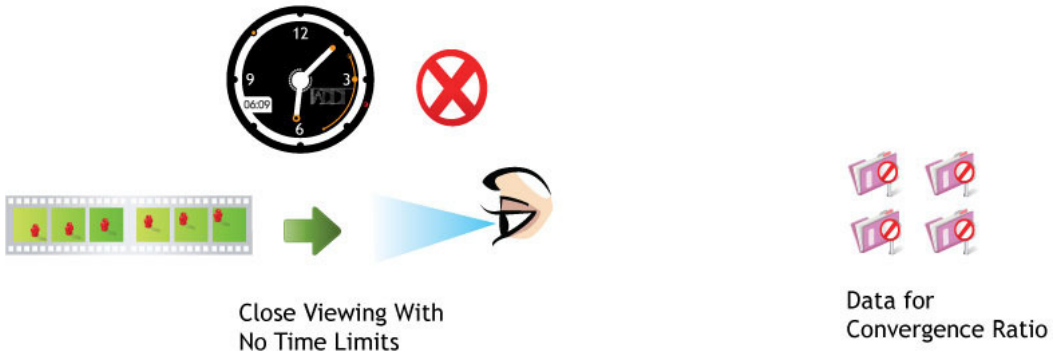


Figure 6-6. Test bench for convergence ratio

Figure 6-7. Test bench for gradient deviation ratio

Figure 6-8 shows the proportion of progressive and interlaced frames detected in both methods. It can be observed that the proportion of the progressive frames have increased in the second method using perception visual examination. The reason for this increase in the proportion of progressive frames is the relation of the human perception with varying temporal and spatial frequencies; the amount of spatial detail human eyes can perceive decreases with an increase in the frame rate and viewing distance.

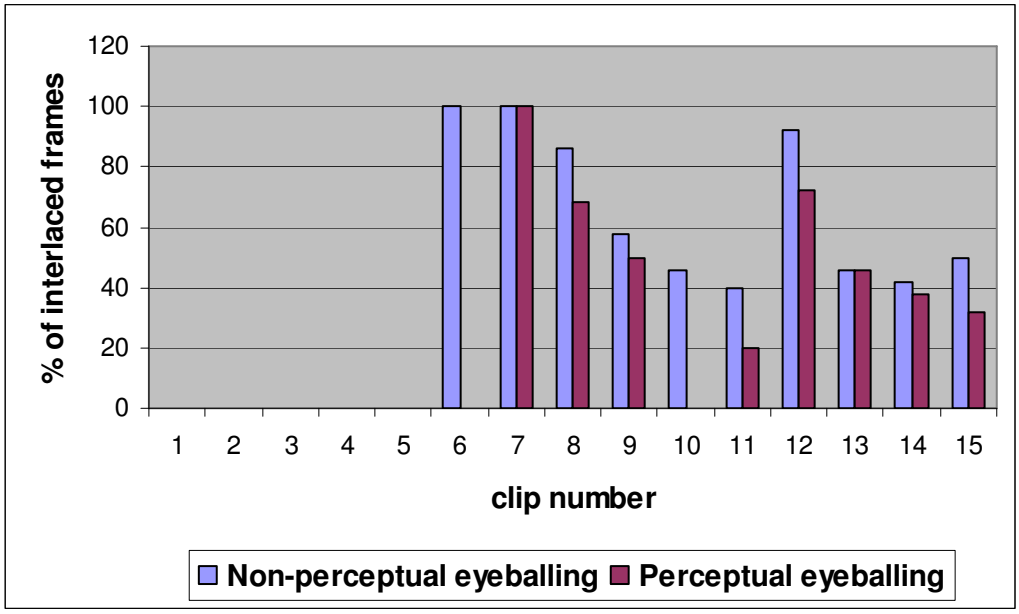


Figure 6-8. Percentage of Interlaced frames

Figure 6-8 also shows that there is no change in the ratio with progressive streams (clips 1-5), but in clips 6 and 10, where the clips are of high spatial detail and exhibit uniform global motion due to camera panning, there is a big difference between perceptual and non-perceptual visual inspection. In contrast, clips 7 and 13, in which the clips are highly interlaced, do not show any variations in the results. These results highlight the relation between human perception, spatial frequency and motion activity explained in section 4-6-2 of Chapter 4.

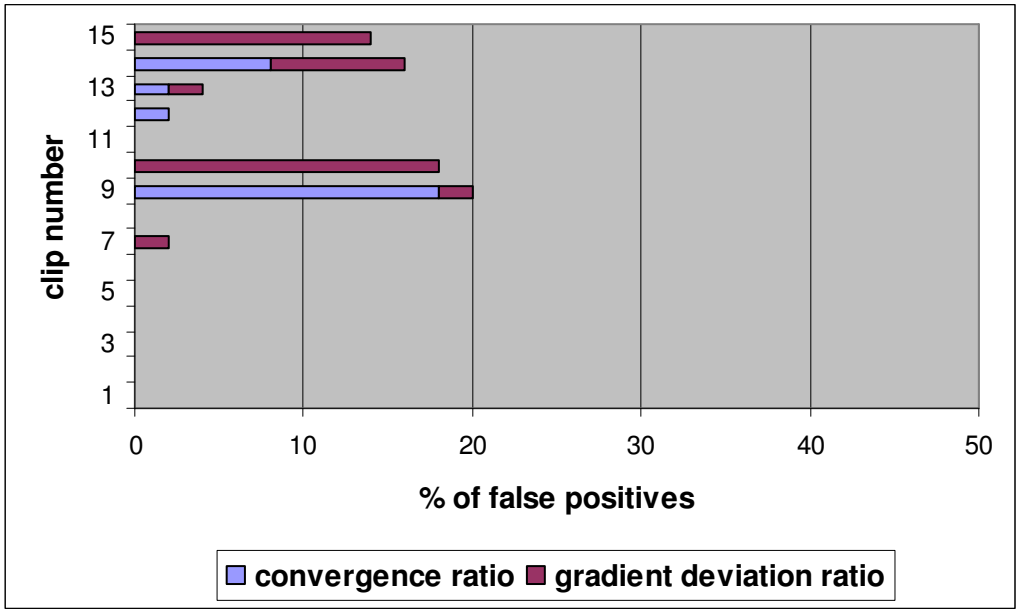


Figure 6-9. False positives generated by metrics

Figure 6-9 shows the false positives generated by both metrics. A false positive may be defined as follows: -

- a progressive frame detected as interlaced
- an interlaced frame detected as progressive.

The false positives are expressed as a percentage. It can be observed from the graph that clips 9 and 14 show the maximum number of false positives generated by both metrics, in which the clips exhibit negligible intensity of combing artefacts. These clips also resulted in ambiguity while classifying the frame as interlaced or progressive during the visual perception process. It can also be observed that the gradient deviation ratio generates some false positives with clips 10 and 15, in which the clips exhibit camera panning and scene change respectively by fading effects. In this case, the motion is globally uniform and does not involve any object based motion; as a result, the mean absolute deviation will be very low due to absence of distinct foreground and background. Hence, the gradient deviation ratio do not reflect the inter-field motion appropriately in these circumstances resulting in false positives, whereas the convergence ratio produces good results. This issue will be dealt with in future research; at the moment this does not significantly influence the precision of the algorithm.

The location of false positives for the convergence ratio metric is graphically shown in Figure 6-10, and similarly, the location of the false positives in the gradient deviation ratio is graphically shown in Figure 6-11. The transition range for the convergence ratio is between 0.40 and 0.45, so the cut-off threshold is set to 0.4250. For the gradient deviation ratio, the threshold is set to 0.5, suitable for a viewing distance of 2 feet. The interlaced to progressive transition occurs at the cut-off threshold. A value less than the cut-off threshold implies that the frame is interlaced and a value greater than the cut-off threshold implies that the frame is progressive. The mathematical theory behind the cut-off threshold value was explained in Chapter 4.

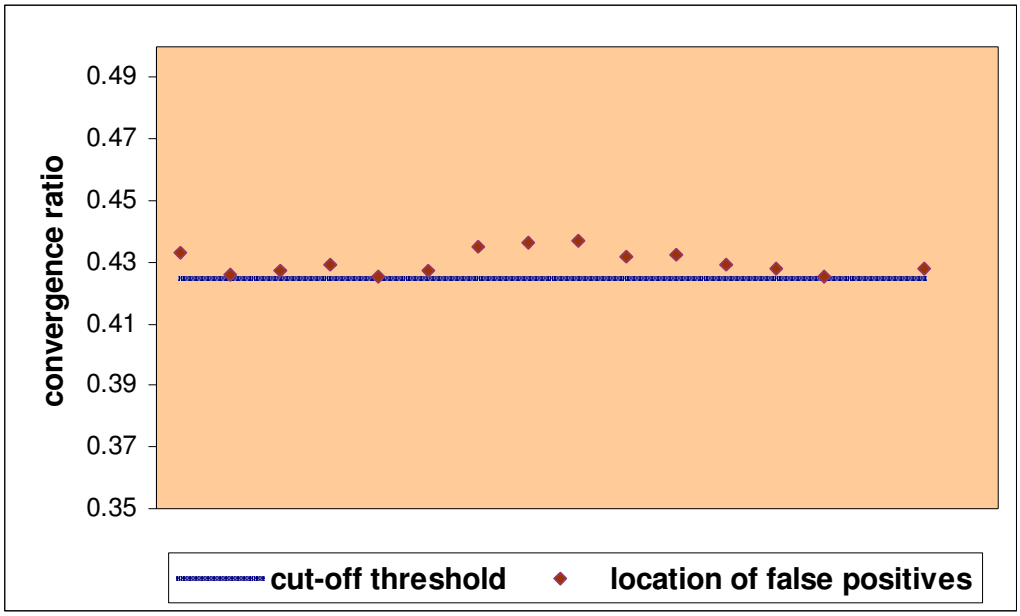


Figure 6-10. Location of false positives for convergence ratio

From the graphs, it is evident that the locations of the false positives are very close to the threshold cut-off. The false positives could have been caused by perceptual errors rather than the metric error. Most of the false positives occur with those frames in which there was some doubt whether the frame should be classified as progressive or interlaced, because of the presence of combing-like artefacts, which were not caused by the inter-field motion. This problem can be mitigated by using ground-truth data in addition to perceptual data. This will be carried out in future research.

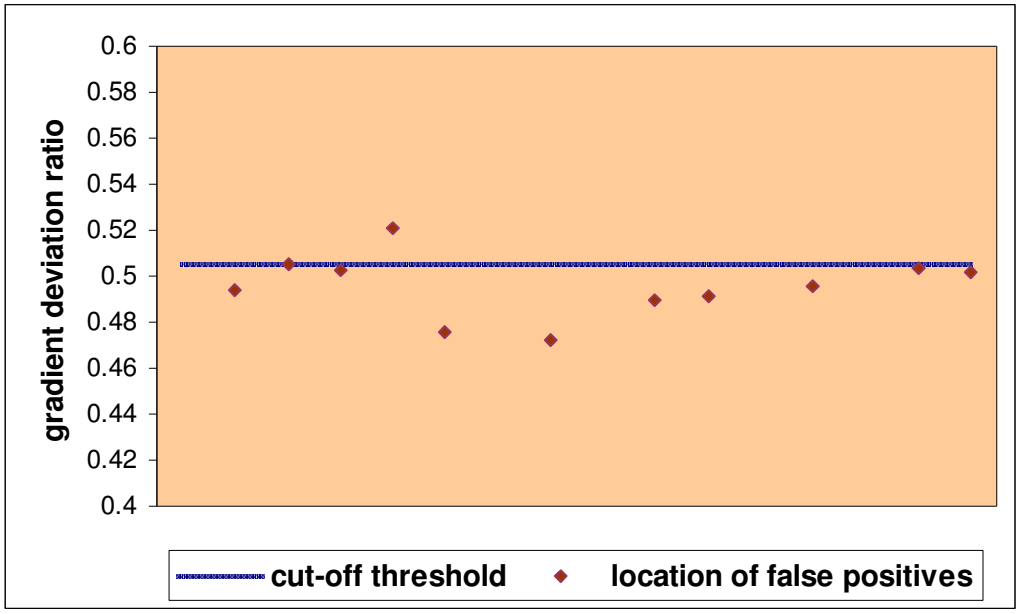


Figure 6-11. Location of false positives for gradient deviation ratio

Figure 6-12 presents the performance comparison of the block-based 'cluster ratio' with proposed frame-based metrics. The cluster ratio is a physical metric not a perceptual metric, so the comparison is performed only with the convergence ratio, which is a frame-based physical metric. Since the cluster ratio operates with an adaptive threshold, the performance of the cluster ratio is compared with the convergence ratio with two different threshold values. Threshold 1 is set to a lower value and threshold 2 is set to a higher value. It can be observed from the graph shown in Figure 6-12, that threshold 1 generates false positives with progressive sequences with high spatial detail, whereas threshold 2 generates false positives with interlaced video sequences, by categorising interlaced frames as progressive. It is observed that the cluster filter, despite offering flexibility for block-based processing, generates false positives due to the adaptive nature of the threshold.

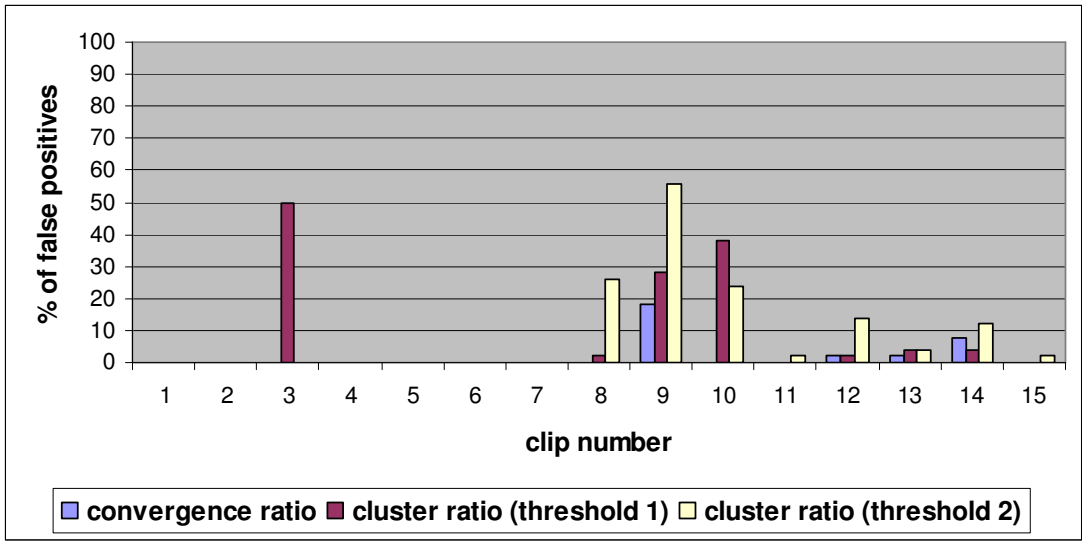


Figure 6-12. False positives generated by cluster filter

6.4.2. Performance Comparison of Proposed Metrics with State of Art Methods

This section compares the proposed new metrics with existing methods. Two methods in existence for interlaced/progressive classification are chosen for comparison, the first method is a simple cumulative difference that represents most methods in existence, which operate on pixel differences. The cumulative difference is calculated by subtracting the fields and performing summation of the absolute difference values. The second method is the linear invariant zipper filter proposed by Baylon and Mckoen (2006), as it is the most efficient metric in existence for interlaced/progressive classification. The threshold for the cumulative difference is set by analysing the data from different video sequences. The threshold for the zipper filter is set, as mentioned, in the patent document (0.05).

The effectiveness of the metric can be assessed by its ability to classify the frames with a stable threshold cut-off. It is observed from the graph shown in Figure 6-13, that the proposed method outperforms existing methods. Existing methods show good performance with progressive videos, as they operate on a very safe margin of error and detect interlaced frames that are very highly interlaced (significant combing artefacts); other frames are tagged as progressive.

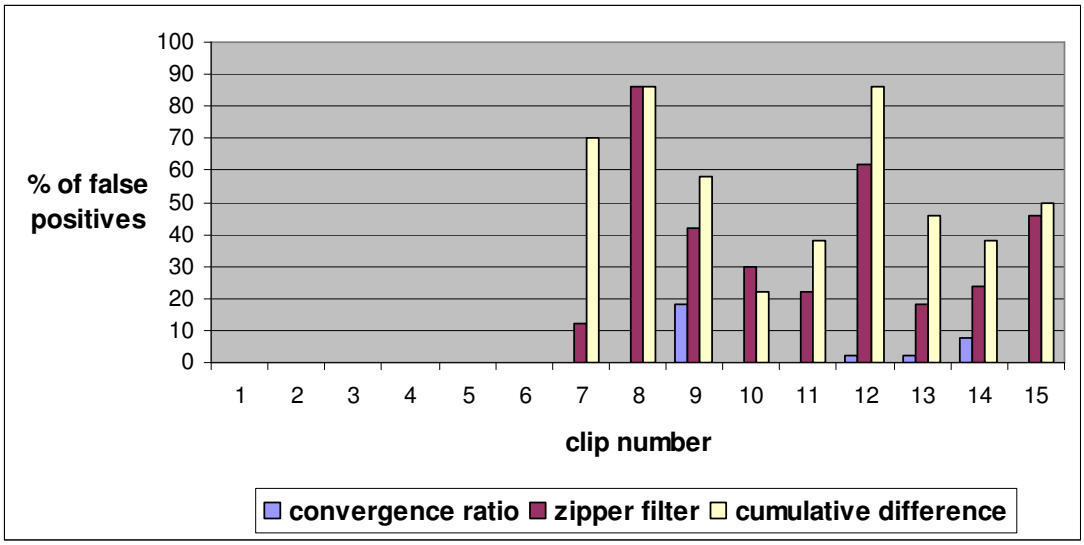


Figure 6-13. False positives generated by various metrics

The reason for such high precision of the proposed metric is that it has been proven in Chapter 4 that below the cut-off threshold, there is no possibility of frames being progressive, and similarly, for the values above the threshold cut-off, there is no possibility of the frames being interlaced. Figure 6-14 illustrates the computation speed of the metrics. It is observed that the convergence ratio has a higher processing time in comparison with the zipper filter and cumulative difference metric. The reason for this is that the convergence ratio is designed to serve as a pre-processing block for many complex algorithms, such as de-interlacing, field reversal and inverse-telecine. The primary operations, like the interpolation and gradient calculations, are already performed in the pre-processing stage of the whole system. This reduces the processing time of the higher layer algorithms. Though it is not fair to compare the proposed metrics with standalone metrics, as they are part of a whole system, the processing time for the convergence ratio is twice that of the time for the zipper filter. However, if the zipper filter is used as a pre-processor module instead of the convergence ratio, the overall processing time of the end-to-end system will be much higher than when using the convergence ratio.

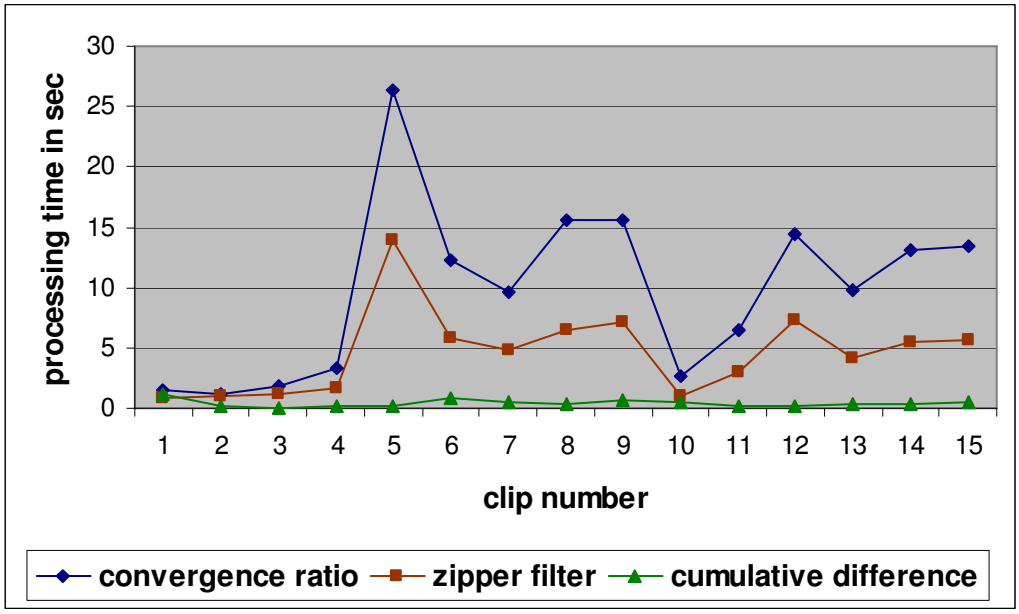


Figure 6-14. Computation speed of the metrics

6.5. Simulation results of Field Reversal and Mixed Pulldown Methods

The same testing architecture used for the inter-field quantifier was extended to test the field reversal and mixed pulldown methods. The proposed correlation algorithms are unique in nature due to integration of both field reversal and mixed pulldown methods in the same module. There are numerous patented pulldown pattern detection methods in existence. Every method is designed for a specific application and utilises many threshold values (magic numbers) for pulldown frame detection; the specific values for the thresholds have not been specified in the documents. Since the proposed method is a generic algorithm that is designed to work with static thresholds, it was difficult to choose the appropriate existing state-of-art methods for comparison. Since the core methodology used in solving both field reversal and mixed pulldown problems is to assign a field order to interlaced frames and not to the pulldown frames, the field reversal algorithm proposed by Baylon and Mckoen (2006) was only chosen for comparison.

Figure 6-15 shows the comparison between existing methods and the proposed method for the detection of field reversal. A false positive may be defined as follows:-

- the field order of an interlaced frames is different from the original field order
- a progressive frame is assigned a field order
- an interlaced frame with inter-field motion is not assigned a field order
- a pulldown frame is identified as interlaced and assigned a field order.

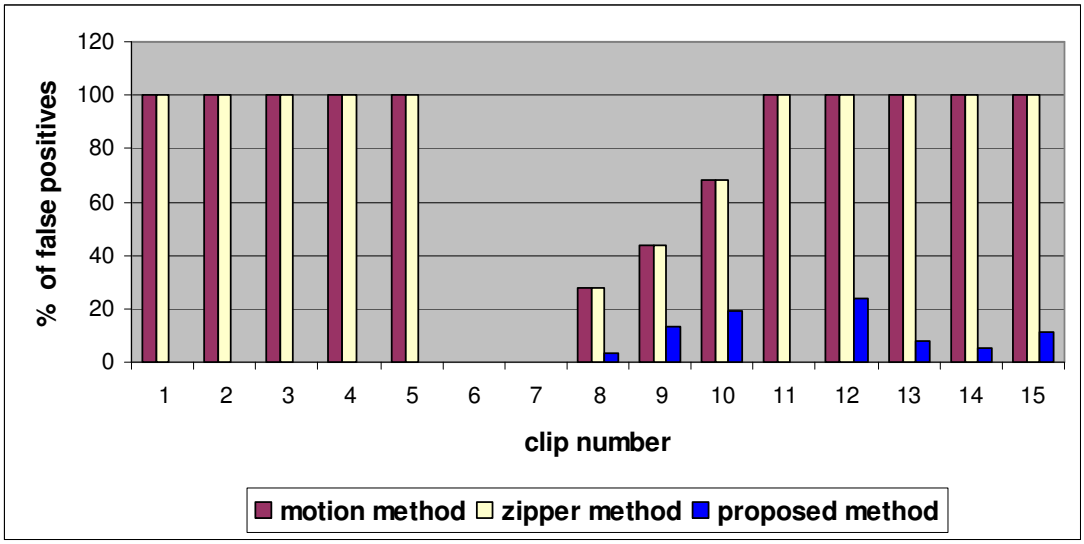


Figure 6-15. Comparison of field reversal methods

The motion correlation method and the zipper correlation method proposed by Baylon and Mckoen (2006) are compared with the proposed new method. The proposed method utilises both PSNR and optical flow as correlation metrics and has a threshold check in place (both thresholds Δ_1 & Δ_2 are set to 0.25). From the graph, it is concluded that the existing methods show good performance with clips 6 and 7, which are highly interlaced and fail with other streams. This is because the algorithms assume that all the video sequences are interlaced and assign a field order to every frame in the sequence, as there is no way of determining the type of the video sequence without using a pre-processor. In the proposed method, the indeterminate results are permitted where the results are inconclusive. If the correlation values are less than the lower threshold value, then the frame is categorised as progressive, as the variation in correlation values might have been due to compression noise. This reduced the number of false positives in the proposed method.

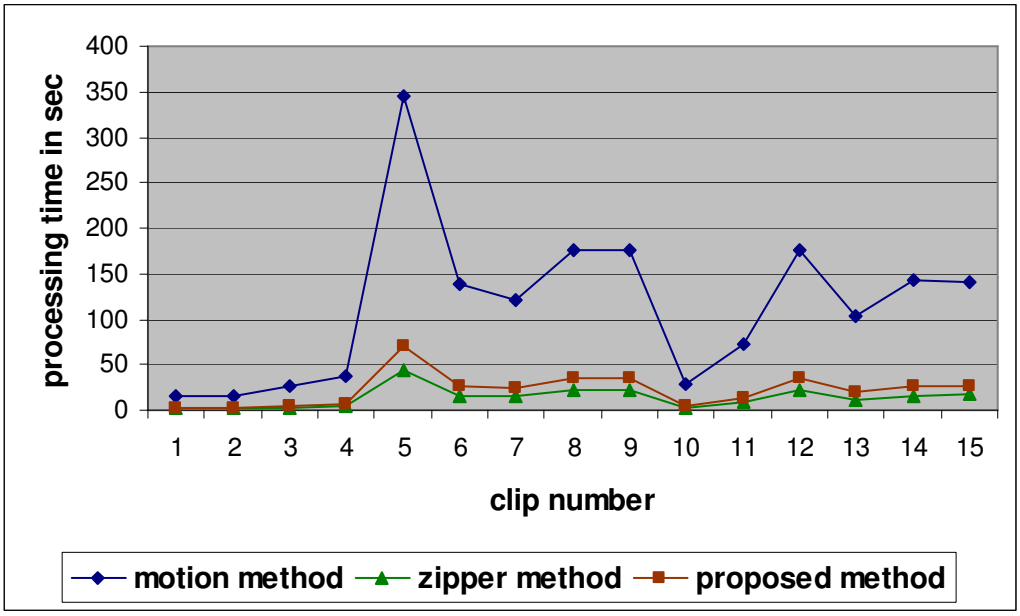


Figure 6-16. Comparison of the computation speed

Figure 6-16 shows the comparison of the processing time of the methods. The motion estimation method is the costliest of all the methods, as it requires more hardware resources. The zipper method and the proposed new method show a similar hardware performance; the proposed method is slightly slower than the zipper method, but considering the number of false positives generated by other methods, the proposed method is very efficient in detecting field reversals and mixed pulldown frames.

It is accepted by both Baylon and Mckoen (2006) and the author that it is not feasible to apply the field reversal method without knowing the nature of the frames in advance. Baylon and Mckeon (2006) proposed the zipper filter to classify a frame as being interlaced or progressive, and applied the motion estimation method to estimate the field order. In the proposed method, the inter-field quantifier (gradient deviation ratio, convergence ratio) designed in Chapter 4 is used to classify a frame as being progressive or interlaced and the field reversal algorithm is applied to determine the field order.

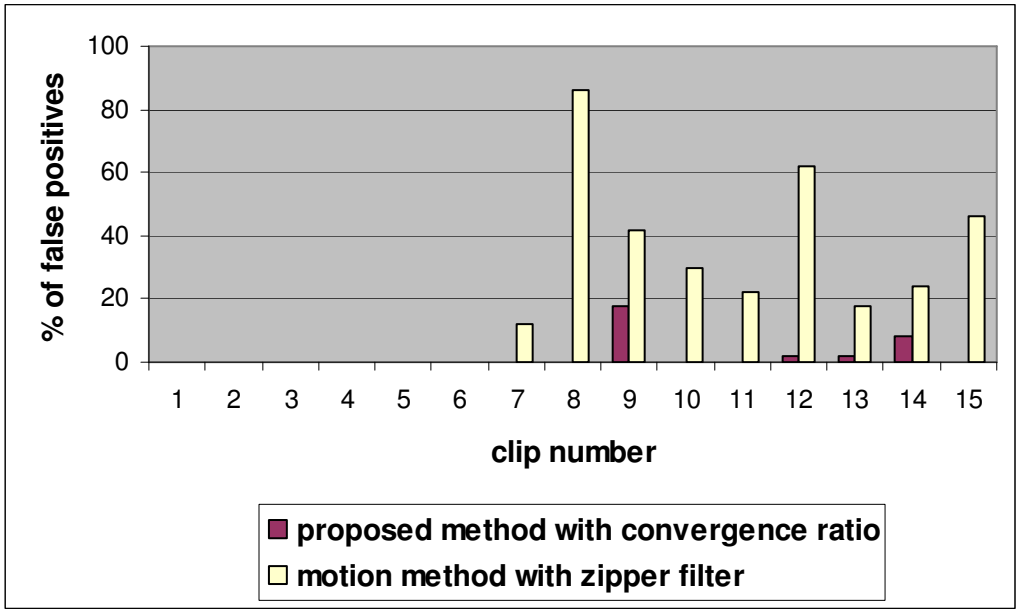


Figure 6-17. Performance of the methods with inter-field quantifier

Figure 6-17 shows the number of false positives generated by the existing motion method with a zipper filter and the proposed new correlation method with the convergence ratio metric. It can be concluded that the percentage of false positives generated by the proposed method is negligible when compared to the existing method. It can also be observed that the percentages of the false positive generated are exactly same as the results presented in Figure 6-13. This is because the inter-field quantifier drives the whole system by signalling which frames should be processed by the higher layer; as a result, a false positive generated by the inter-field quantifier will reflect as a false positive in field reversal and mixed pulldown detection algorithms. This highlights the need for a powerful inter-field quantifier, as the quantifier's accuracy plays a major role in the performance of the system.

Figure 6-18 compares the processing time of the two methods with a pre-processor. The graph shows that the proposed system outperforms the existing methods with interlaced video sequences, and shows little deviation in performance with progressive video clips. This is because, with progressive clips, the field reversal algorithm is not applied, as the inter-field quantifier indicates the frames to be progressive. In interlaced video sequences, the existing method proceeded with the motion estimation method, whereas the proposed method proceeded with the correlation method, which is indicated in the graph. With the progressive sequence,

the processing terminates with the pre-processor stage and the values are the same as the ones presented in Figure 6-16.

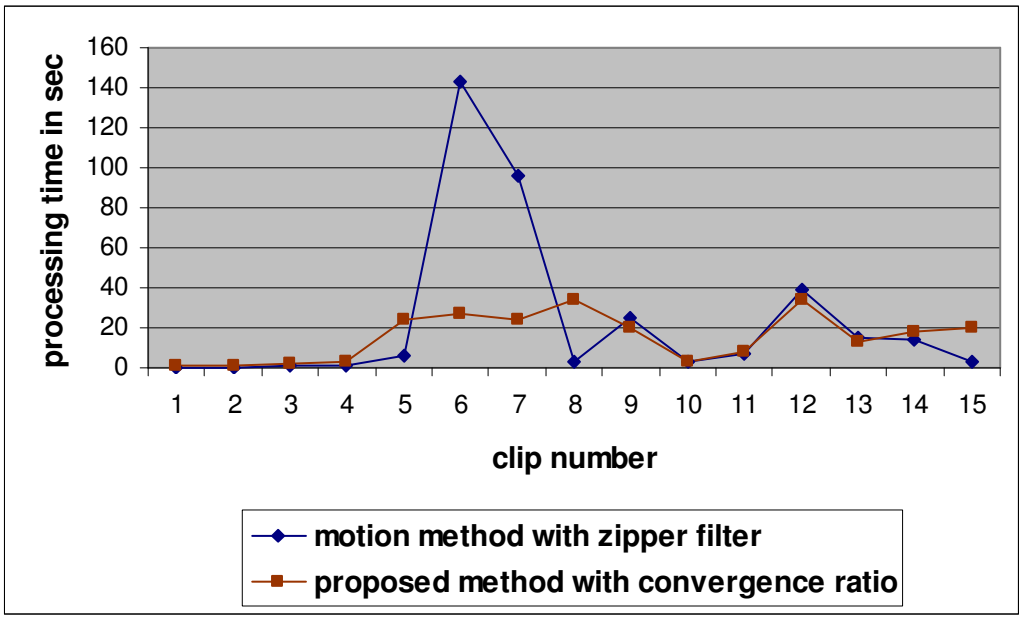


Figure 6-18. Computation speed of the methods with pre-processor

Figure 6-19 compares the proposed method with and without the inter-field quantifier. It can be observed that using the inter-field quantifier causes a significant reduction in the processing time, particularly with the progressive video streams. The improvement in the processing time with the usage of an inter-field quantifier with the interlaced video sequences is negligible.

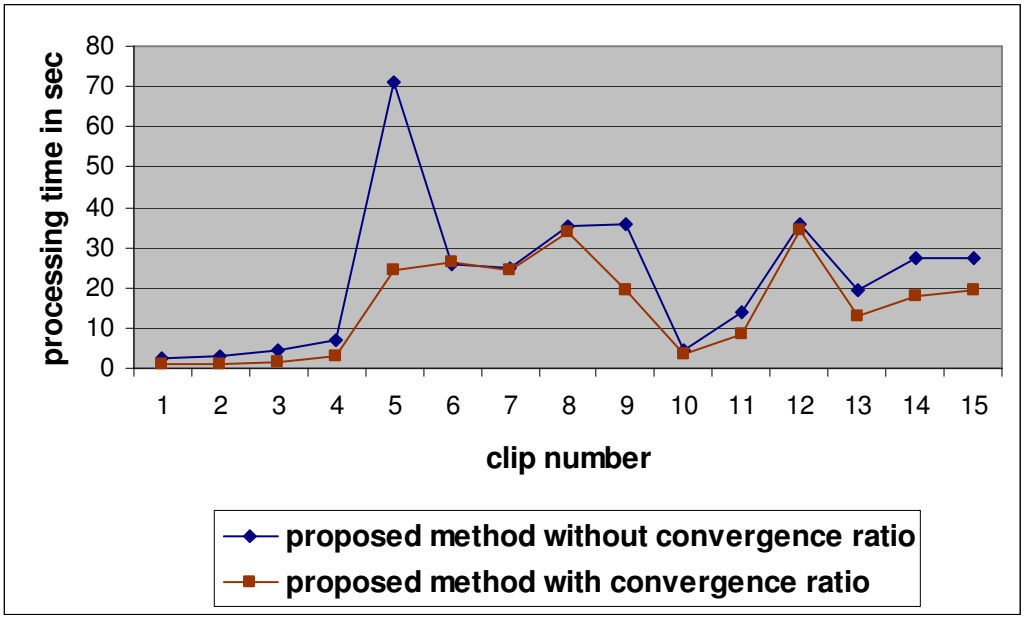


Figure 6-19. Computation speed, with and without using the convergence ratio

Figure 6-20 shows the comparison of existing methods with the proposed methods with and without block-based processing. The block-based processing segments the frame into multiple blocks and the cluster filter is used to determine the presence of inter-field motion. Subsequently, if inter-field motion is present, then the block is processed for field reversal and mixed pulldown algorithms.

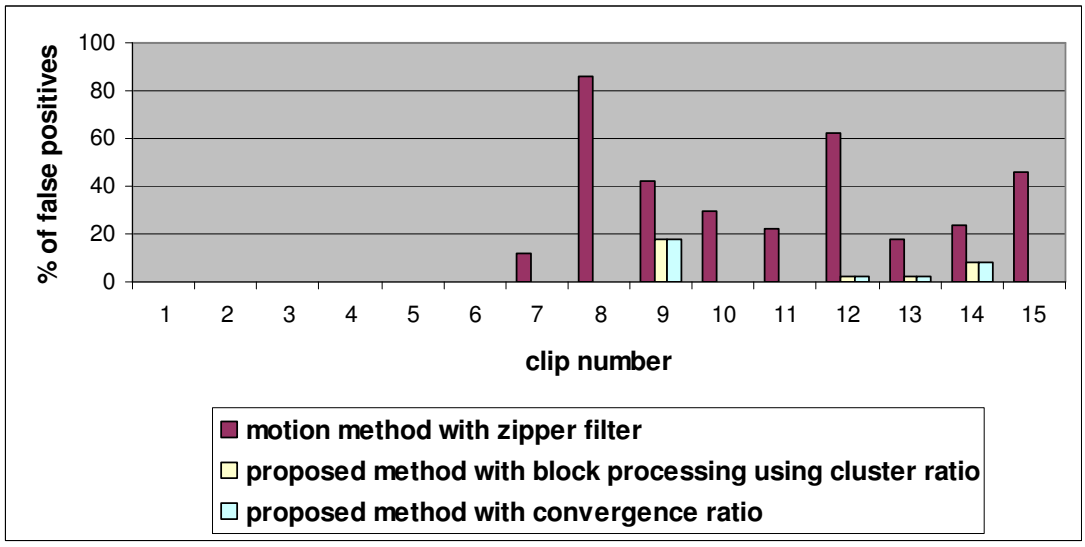


Figure 6-20. False positives with various pre-processors

It can be observed from the graph shown in Figure 6-20 that the percentage of false positives generated by the block-based processing is the same as that of frame-based processing, but the real advantage is with the reduced processing time.

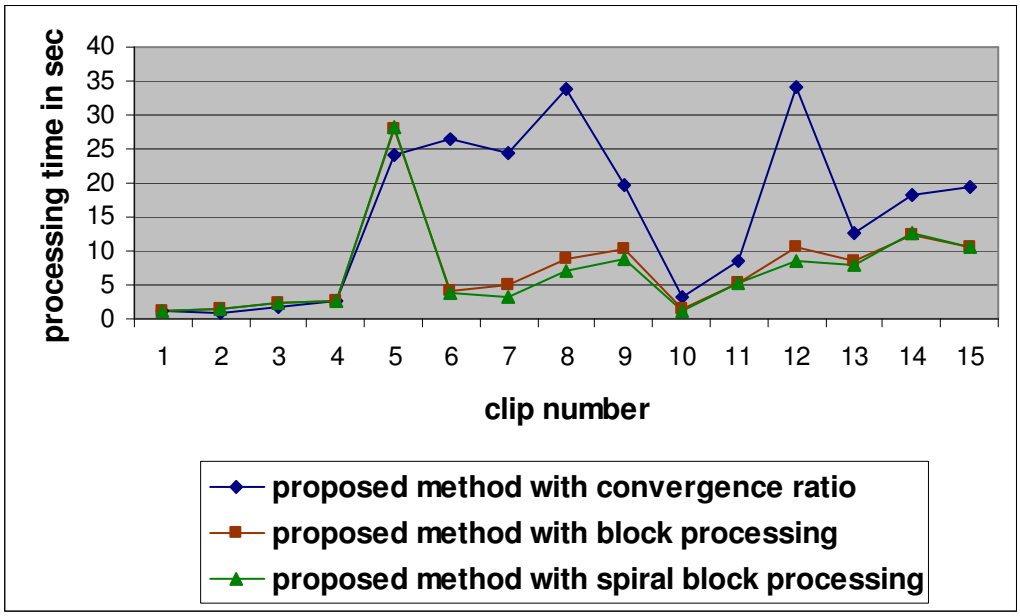


Figure 6-21. Computation speed of frame and block-based processing methods

Figure 6-21 shows that the block-based processing shows a significant reduction in the processing time with the interlaced video sequences and no change in the processing time with the progressive sequence. The reason for this is that the basic principle behind block-based processing is to find the blocks that have inter-field motion within a few iterations, but in a progressive sequence, none of the blocks will have inter-field motion and this will result in all the blocks in the frame being processed, which is equivalent to frame-based processing.

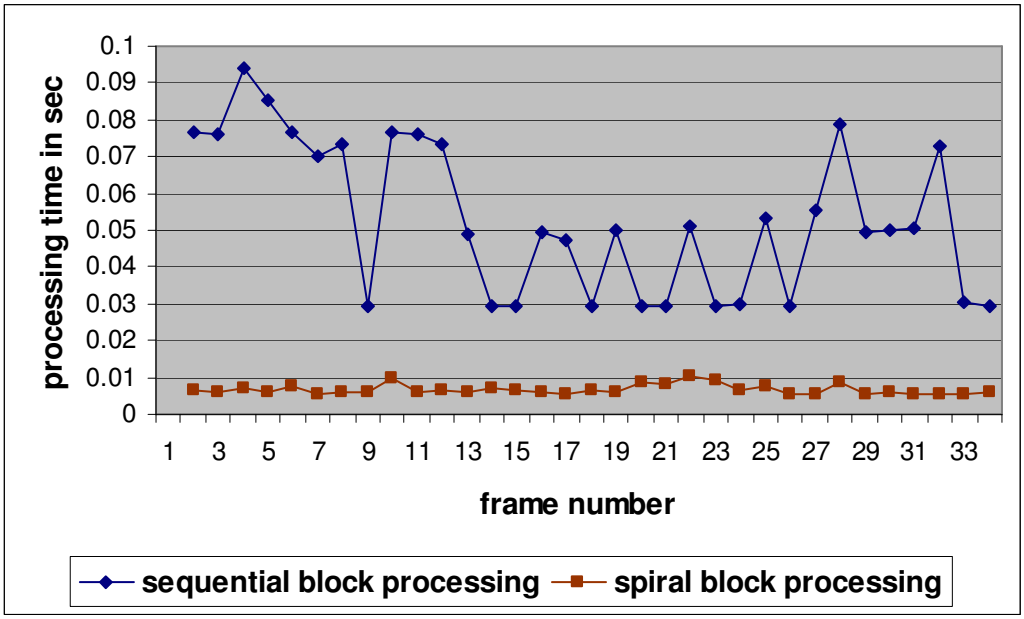


Figure 6-22. Sequential and spiral block processing on an interlaced sequence

Figure 6-22 shows how the processing time differs on an interlaced video stream with different block-based processing methods. The video sequence, 'TEK_Guards_Bottomfield', which is highly interlaced, is used in the simulations. It can be observed from graph that the sequential block processing, which starts processing the blocks from the left corner of the image, is much slower than the spiral-based processing, in which the blocks are processed starting from the centre and spiralling out. A significant saving in the processing time is achieved by using the spiral method, as there is a higher probability of capturing the blocks with inter-field motion with minimal iterations. However, in contrast, Figure 6-23 shows no change in the processing time when applied to a progressive video sequence, 'TEK_Women_Progressive', as none of the methods will find a block with inter-field motion and so all the blocks will end up being processed.

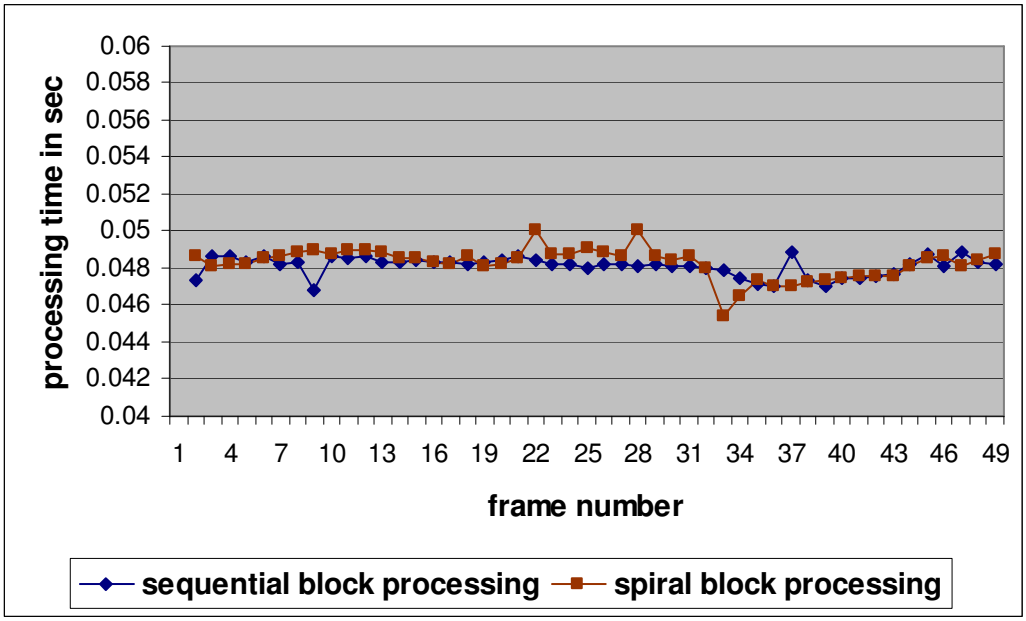


Figure 6-23. Sequential and spiral block processing on a progressive sequence

6.6. Real Time Testing of the Algorithms

The algorithms were integrated with Cerify equipment explained in section 6.2 after rigorous testing with more than 100 clips with varying characteristics. The initial development used convergence and gradient deviation ratio for quantifying inter-field motion, but was later replaced by cluster filter due to constraints on speed. The spiral search method used in Matlab prototype was replaced by strip search method in the real-time version. For field reversal, the user interface offered options to choose a specific field order (top_first, bottom_first) and location of the clip to be processed. Similarly, for mixed pulldown detection, the user interface offered options to choose a specific video type (interlaced, progressive, uniform pulldown and mixed pulldown) and location of the clip to be processed. Cerify processed the clip in accordance with the options chosen and generates an error log with the frame number and the type of the error detected in the frame.

The trial test run was carried out at MTV. Music videos that exhibited visual artefacts were fed into Cerify and processed. Cerify detected field reversal and mixed pulldown errors in numerous video clips with good precision. The trial testing was considered a complete success.

6.7. Summary

The performance evaluation is presented in Matlab, but in real-time, the C++ code performed approximately 100 times faster than the simulation setup, which makes it suitable for real-time implementation. The significance of the research is that the development process was performed in a commercial environment, not in a laboratory environment. The performance results of convergence, gradient deviation and cluster ratios show the different flavours of the inter-field quantifier and how they may be used in different situations based on the requirements. The performance results of field reversal and mixed pulldown algorithms show the step-by-step process of reducing the complexity of the whole system without influencing the quality of the results. The contribution of the inter-field quantifier to the precision of field reversal and pulldown algorithms highlights the extensive design process undergone in Chapter 4. The results also show that the algorithms have been optimised for maximum hardware performance, which gives the flexibility of implementation either as ‘offline’ or ‘on-the-fly’ processing system.

7. A Critical Investigation on Channel Coding and Source Error Resilience

The previous chapters investigated the problems occurring in the editing layer due to human errors, and solutions for the problems were proposed in the source coding domain. This chapter investigates the errors occurring in the channel during transmission, which is beyond human control. Both mobile channels and video transmission differ in characteristics from traditional Gaussian wireless channels and nature of data respectively. Since the characteristics of the underlying transmission deviate from the normal behaviour, the protocol layers designed for handling traditional communication may not function properly. Aforementioned issues and the broad definition of error protection proposed by Shannon in his seminal work are the primary motivations behind this chapter.

7.1. Introduction

This chapter investigates the question: "Is channel coding the best way of providing error control for video bitstreams?" It is argued in this chapter that channel coding, which is primarily designed for counteracting bit errors in a data stream where the bits are statistically independent from the application layer perspective may not be the best option for video streams where the bits are statistically dependent. The result of the discussion strengthens the main hypothesis further, as the flexibility of the source coding layer is justified. One of the primary reasons why the channel coding is preferred over source coding for error protection is that the information theory concepts can be applied easily on the channel coding layer. This chapter presents the concepts of information theory from the perspective of the blocks rather than the bits and argues that the information theory concepts could be also applied in the source coding layer. The chapter ends with the conclusion that by using some non-traditional methods of error control, the following secondary hypothesis holds true: "If the channel is characterised by high packet loss rate of a bursty nature, the optimum quality of multimedia transmission can be achieved by

assigning redundancy bits for error control to the source coding layer rather than to the channel coding layer". Finally, the aim of the chapter is not to challenge the channel code's lack of suitability for mobile multimedia communications; it is about approaching the problem logically with the intention of identifying potentially a better way of coding.

This chapter presents a very broad discussion of the pros and cons of providing error resilience from the channel coding and source coding layers. In section 7.3, the error propagation characteristics of the video stream are presented along with the investigation into variations in the interpretation of information theory concepts when approached from the different layers of protocol stack. In section 7.4, the above discussion is extended to advanced information theory concepts and useful interpretations are presented. In section 7.5, an experimental setup is explained to confirm that Shannon's theorem holds true in the source coding layer. In section 7.6, a critical review of the results from the previous section is presented and some data hiding methods that are under development are explained. Section 7.7 concludes the chapter with a summary of all the investigations.

7.2. Bits Error Ratio vs Block Error Ratio vs Packet Error Ratio

Traditionally, the errors occurring in the channel coding domain are measured using the Bit Error Ratio. The Bit Error Ratio (BER) is the number of bits in the received bitstream that have changed during the journey across the channel. When the channel codes are used for adding additional redundancy to detect and correct errors, in line with Shannon's theorem (Shannon, 1948), the BER does not reflect the error performance of the received bitstream. The term Block Error Ratio (BLER) is used as a measurement metric; this is the number of blocks that contain at least one bit error. The blocks are primary data units of any communication protocol, and can be of different sizes based on the standard. The blocks are fed into the channel coding layer for the addition of redundant bits for error correction. The bit errors occurring in the block can be corrected in the receiver to a certain extent using the extra redundancy.

In advanced communication protocols, the definition of a 'block' is used interchangeably with the term 'packet' and differs depending on the data link layer of the underlying communication protocol; for example, in wireless systems, one RLC (Radio Link Unit) unit is known as a block. In other words, a block may be defined as a meaningful combination of bits as described by the underlying protocol. The graphical illustration of the generic architecture of a communication protocol is shown in Figure 7-1.

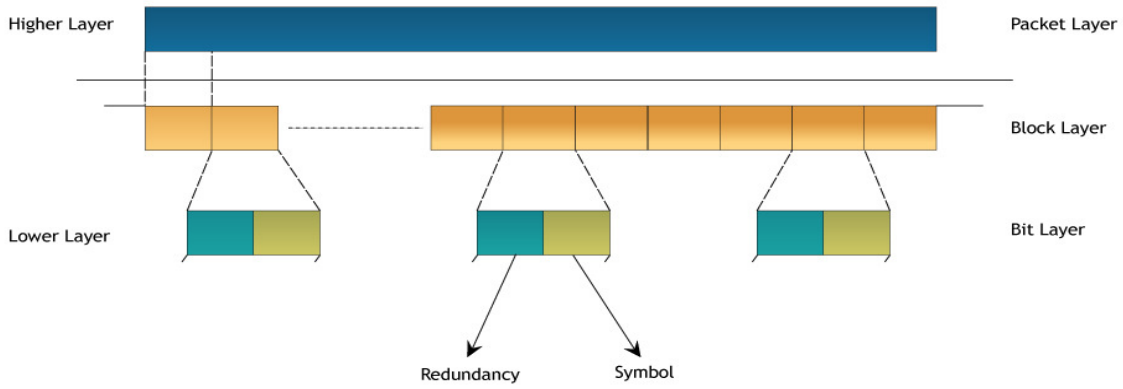


Figure 7-1. Bits, blocks and packets

The block and packet error ratios will reflect the loss in the higher layer reasonably well if the data are statistically independent (web, text and so on). In this specific research, the nature of the application is video, in which the basic unit is a video block. A video block is defined as a combination of multiple macro-blocks, which represents a meaningful segment in video coding context.. This is identified as a major problem with the wireless multimedia transmission, as a successful decoding of a video block can occur only when all the blocks constituting the video block are received successfully. This problem has been addressed by many researchers (Kurceren and Modestino, 2000). Even if the error propagation effects are ignored, the block and packet error ratios will reflect the physical impact on the higher layer reasonably well only if a video block is completely contained within a block or a packet. This is very unlikely, as the video compression generates variable length blocks and the communication protocols change the block length dynamically based on the channel constraints (SNR, feedback and so on).

This limitation of metrics establishes that the bit, block and packet error ratios will not accurately quantify the error impact at the higher layers of the protocol stack. The channel coding algorithms that claim performance increase using these metrics are inefficient, as they will not reflect the perceptual impact of the error because they operate on the lower layers of the protocol stack. Significant improvements have been observed in operating at the higher layers of the protocol stack in terms of delay (Lo et al., 2005), quality (Qu et al., 2004) and flexibility (Pathak et al., 2005). This chapter explains how errors affect the quality of the decoded video because of the lack of transparency among the layers of the protocol stack, and subsequently, it is demonstrated that a better performance can be achieved by providing error protection from the higher layers of the protocol stack.

7.3. Error Propagation in Compressed Video Streams

The mobile multimedia system is driven by the property of 'statistical dependency'. The statistical dependency occurs in different layers due to the following reasons:-

- variable length codes in the spatial domain
- prediction in the temporal domain
- bursty errors in the channel domain

This section investigates each issue in detail from the perspective of the channel and the source coding layer.

7.3.1. Issue Due to Variable Length Coding

The unit of information is the bit; if the probability of occurrence of a symbol is less, then the information carried by it is more and vice versa. The advantage of using information theory is to avoid redundancy in the data transmission. The redundancy normally occurs when the probability of the symbols emitted from the source are not equi-probable. The symbols with a high probability are apparently more likely to be transmitted repeatedly, and should be represented by a shorter codeword than that of the symbols with a lower probability. The process of variable length coding utilised in the compression models is based on the above principle.

The advantage of using variable length coding instead of fixed length coding will disappear if the probabilities of the occurrence of different symbols are the same.

When a block of video data is represented using the variable length codes, the code words become statistically dependent and can only be decoded sequentially. If there is an error in the bitstream, the decoder cannot continue decoding subsequent code words and this will result in data being dropped until the boundary of the next video block. When a block of video data is distributed across multiple packets, one bit error in one of the packets implies that all other packets containing data from the same video block will become meaningless (Zhang et al., 2008). The packet error ratio undervalues the damage caused in the higher layer. This phenomenon is illustrated in Figure 7-2.

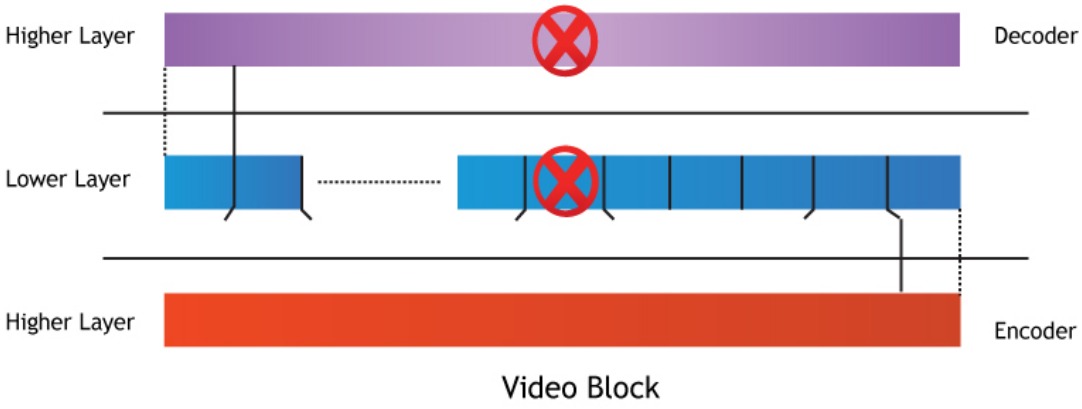


Figure 7-2. Spatial error propagation

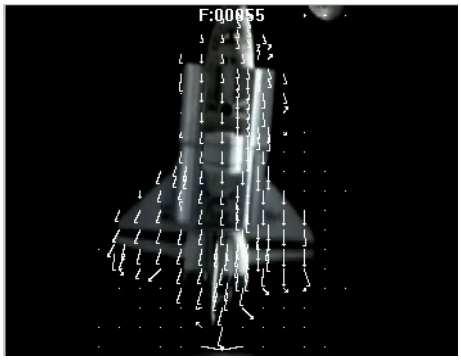
The parameters like boundary, size and importance of the contents of the video packet are visible only from the source coding layer. This makes the source coding layer a better platform than the channel coding layer, in which the video data are visible only as a lengthy string of bits. This section explained the spatial propagation of the errors due to the variable length codes; the next section explains the temporal propagation of the errors due to prediction.

7.3.2. Error Propagation due to Prediction

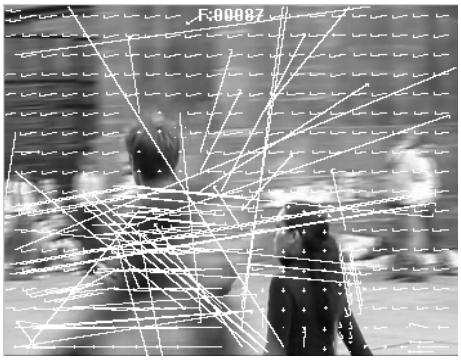
If the block of data is erroneous due to the corruption of variable length codes, then the impact on the current frame is spatial. The spatial impact might become temporal and reduce the perceptual quality of the succeeding video frames, despite receiving the successive blocks without any errors. This is because of the prediction principle used in motion estimation and the compensation modules of the video compression system. The condition for the successful reception of a block in a frame will depend on the successful reception of the block in the corresponding position in the previous frame. Due to the predictive nature of the coding, the blocks become statistically dependent.

The probability of error is based on the volume of bits in the lower layers, but its true reflection in the higher layer will depend upon its location of impact. The statement, 'impact of the error on the higher layer can only be projected by understanding its location' is explained in detail. The displacement of the macro-blocks within the frame is unpredictable, as each video stream has different motion characteristics. On analysis, through multiple simulations, it has been found that the motion pattern of a video frame can be categorised into one of the three types listed below: -

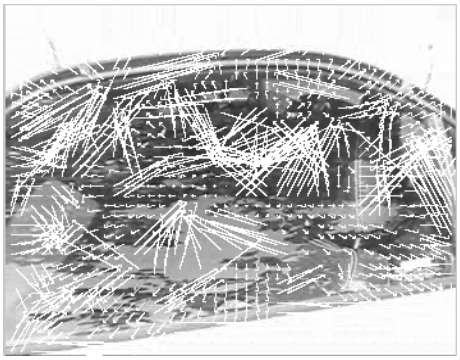
- uniform displacement (Figure 7-3.a)
- cluster based displacement (Figure 7-3.b)
- random displacement (Figure 7-3.c).



(a)



(b)



(c)

Figure 7-3. Motion patterns [Source: Tektronix MTS4EA Analyzer]

In uniform displacement, all the macro-blocks move more or less in the same direction (object moving along a stationary background). In cluster-based displacement, a specific group of macro-blocks will follow a uniform displacement (multiple objects moving in different directions). In random displacement, the macro-blocks exhibit random characteristics and do not show any correlation among themselves (zooming or focussing).

An error occuring in the video stream will exhibit different propagation characteristics based on the displacement characteristics explained above. The error propagation can be generally classified into three different types: -

- short time propagation
- slowly fading propagation

- exponential propagation.

In 'short time propagation', errors do not propagate to more than a frame because of a null motion vector or corresponding block in the successive frame being replaced by an intra macro-block by the AIR (Adaptive Intra Refresh) method. This scenario could be modelled by the binomial distribution; 'p' is the probability of success (frame with error), 'n' is number of frames or trials. Both 'p' and 'q'(1-p) will be 0.5 (7-3-1).

$$b(k;n,p)={}^{n}C_{k}.p^{k}.q^{n-k} \qquad k=0,1,2,.....n \qquad (7\text{-}3\text{-}1)$$

In 'slowly fading propagation', the error propagation rapidly increases to a level and slowly reduces to minimum. This scenario could be represented by a Gaussian or Normal distribution (7-3-2). The intensity of the fade and the shape of the curve are determined by μ and σ, which correspond to the distortion and the rate of insertion of AIR (Adaptive Intra Refresh) blocks respectively.

$$f(x)=\frac{1}{\sqrt{2\pi\sigma^{2}}}.e^{\frac{-(x-\mu)^{2}}{2\sigma^{2}}} \qquad (7\text{-}3\text{-}2)$$

With exponential propagation, the influence of error propagation increases with time and can be represented by the exponential distribution.

$$f(x)=\mu e^{\mu x} \qquad \begin{matrix} x>0 \\ 0, otherwise \end{matrix} \qquad (7\text{-}3\text{-}3)$$

If there were extensive motion between the frames, a minor information loss would lead to exponential error propagation on a large scale. The definition of extensive motion will be one of the scenarios mentioned below: -

- non-uniform object moving across the frame
- rapid camera panning and zooming
- fast appearance and disappearance of object occupying more than 50% of the area of the frame

- multiple objects moving in different directions rapidly across the frame.

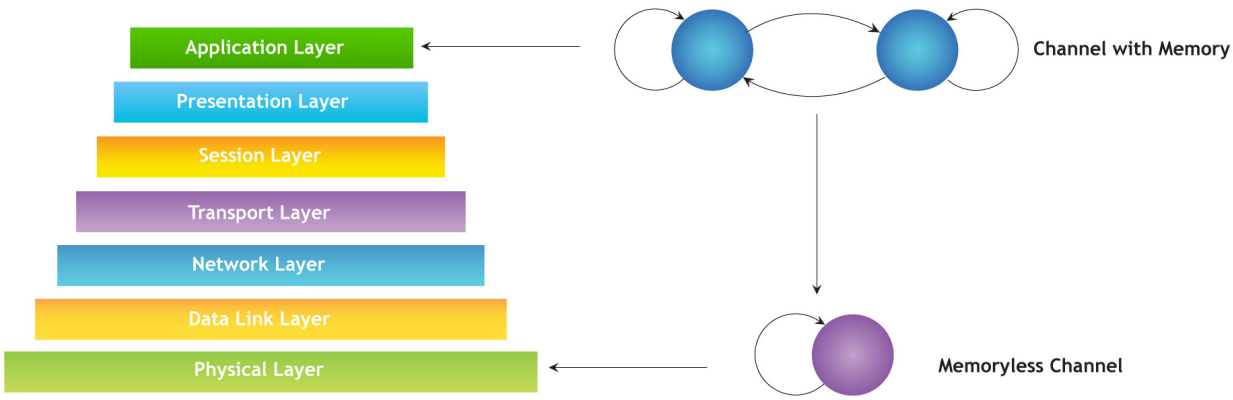


Figure 7-4. Dynamic channel variation in protocol stack

The status of a video block at any point in time depends on the neighbouring spatial and temporal blocks. When the video bitstream is seen from the lower layer of the protocol stack, the channel is memory-less; as we proceed to the top of the protocol stack, the channel holds some memory. This issue is illustrated in Figure 7-4. This establishes the fact that the channel coding schemes operating on the lower layers with an assumption that the channel is memory-less will not perform well when the application in the higher layer holds some memory.

This problem of layer variation can be effectively addressed by projecting the impact of a corrupted video block in the spatial and temporal domain. The error protection can be varied based on the importance of the video block. This method is known as unequal error protection and has been addressed by many researchers (Chen et al., 2007; Thomos et al., 2006). The importance of a video block to the overall frame quality is visible only from the source coding layer, which again shows the flexibility of the source coding layer in the channel coding layer.

7.3.3. Impact of Bursty Errors on a Data Dependent Application

Extending the discussion on the source entropy from section 7.3.1, errorless transmission is expected to occur when the transmitted symbol entropy is the same as

the received symbol entropy. This is very unlikely in existing communication networks, as most of the time, the received symbol is not identical to the transmitted symbol due to channel errors. The probability of a source symbol being corrupted during transmission along the channel can be represented by the channel matrix. Equation (7-3-4) shows the channel matrix; if the diagonal elements are one, then it implies that there is no uncertainty in the channel.

$$Channel\ matrix = P\left(\frac{O}{I}\right) = \begin{bmatrix} P\left(\frac{o_1}{i_1}\right) & P\left(\frac{o_2}{i_1}\right) & ---- & P\left(\frac{o_n}{i_1}\right) \\ P\left(\frac{o_1}{i_2}\right) & P\left(\frac{o_2}{i_2}\right) & ---- & P\left(\frac{o_n}{i_2}\right) \\ - & - & & - \\ P\left(\frac{o_1}{i_n}\right) & P\left(\frac{o_2}{i_n}\right) & ---- & P\left(\frac{o_n}{i_n}\right) \end{bmatrix} \quad (7\text{-}3\text{-}4)$$

The channel matrix depicts the level of uncertainty added by the channel to the source symbols. The probability at each position gives the value of uncertainty in identifying a received symbol as a corresponding source symbol, in probability terms; it is the conditional probability of receiving a symbol successfully on transmission of a source symbol. This channel matrix plays a very important role in designing the communication systems.

$$Information\_Rate = \left(H\ (I) - H\left(\frac{I}{O}\right) \right) \times Symbol\_Rate \quad (7\text{-}3\text{-}5)$$

$$Maximum\_Information \mapsto\ H\left(\frac{I}{O}\right) = 0 \quad (7\text{-}3\text{-}6)$$

$$No\_Information \mapsto\ H\left(\frac{I}{O}\right) = H(I) \quad (7\text{-}3\text{-}7)$$

Equation (7-3-5) shows that the information rate depends on the source probability and the channel uncertainty. The maximum information is transferred across the channel when there is no uncertainty in the channel (7-3-6), whereas no information is transferred when the uncertainty imposed by the channel is as big as the source entropy (7-3-7). One of the important aspects of the channel error

probability is that it is a function of the channel noise probability distribution and the modulation technique used for transmission. The channel matrix reflects the resilience of a modulation technique against the channel uncertainties.

It is established that the turbo codes are more powerful than the convolutional codes in terms of error correction. The unique aspect of the turbo code is that it can operate at near Shannon's limit, as the data rate can be increased without increasing the transmit power. One of the reasons why the turbo codes outperform other coding methods is that this method combines the two channel coding methods and operates as a concatenated coding scheme. To achieve the same performance using a single channel code, the code words should be very long (Kurceren and Modestino, 2000).

The main drawback of the turbo coding and convolutional coding is that it has not been designed for bursty channels; it achieves maximum performance in random channels. The bursty errors must be transformed into random errors by interleaving. The nature and type of interleaving plays a major part in the turbo code's performance (Tepe and Anderson, 2001). The turbo code's performance degrades in fading channels if the interleaver is not used.

The property of the bursty error is that the errors are imposed in clusters, so if the burst size is 100 bits, it will wipe out a string of bits. Assuming an average video block size of 200 bits, the errors will be contained within a block, if the data are not interleaved. The effect of one bit error in a video block is the same as multiple errors, provided the errors are confined within the video block boundary. In the channel coding layer, the data must be interleaved, so that the error clusters are disintegrated into random errors for the convenience of the channel coding algorithms. If the channel coding algorithms fail to correct the random errors, this effectively increases the damage by distributing the errors across multiple video blocks; the errors could have been confined with a video block in the source coding layer. This is graphically illustrated in Figure 7-5.

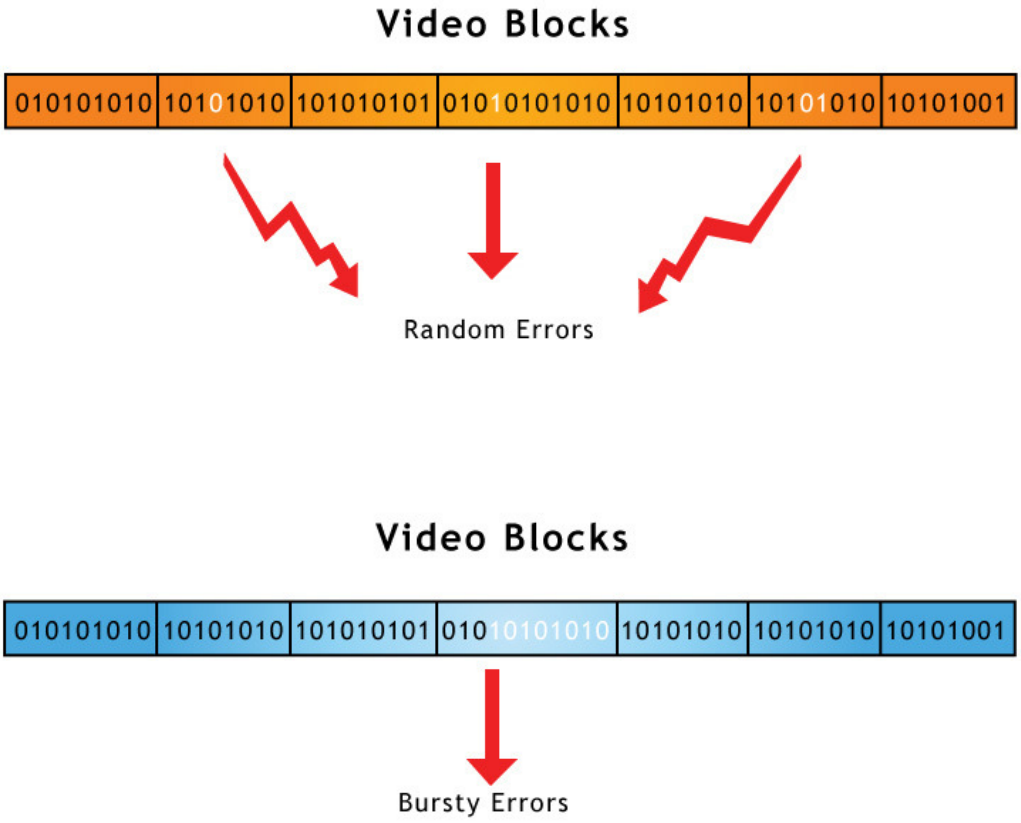


Figure 7-5. Impact of random and bursty errors on video blocks

The discussions presented in this section can be summarised as: -

- The error characteristics of the lower and higher layers may vary depending upon the application data.
- The decision on the nature of the channel must be determined by examining both the lowest and highest layers of the protocol stack.
- Memory-less channel cannot be assumed when the target application is data dependent.
- The basic unit of information theory can be considered to be valid only if the application layer can understand the unit independently.
- In a data dependent application like digital video, the impact of an error must be quantified by higher layer objective metrics like PSNR, which reflect the true impact of the error, rather than using volumetric metrics.

7.4. Advanced Information Theory Principles

This section examines advanced information theory concepts. The information theory concepts are reviewed from the channel coding layer, in which the fundamental meaningful unit is a 'symbol', and subsequently, the concepts are reviewed from the source coding layer, in which the fundamental meaningful unit is a 'video block'. An important attribute used in the information theory is 'mutual information'. The mutual information represents the amount of information conveyed by the source and received symbols about each other. In other words, the mutual information represents the correlation between the source and received symbols. If the mutual information or correlation is high, then there is a high possibility that the source symbol will be predicted correctly from the received symbol, as there exists a statistical dependence between the symbols. If there is no correlation, or if the mutual information is zero, then the source and received symbols are statistically independent, so the received symbol will not reflect the source symbol in any way. The mutual information is represented by the joint probability distribution, source entropy and received symbol entropy. The joint probability will depend on the source and received symbol entropies and the channel matrix. Equations (7-4-1, 7-4-2 and 7-4-3) show the mathematical relationship between the above explained parameters.

.

$$H(I,O) = H\left(I/O\right) + H(O) \tag{7-4-1}$$

$$H(I,O) = H\left(O/I\right) + H(I) \tag{7-4-2}$$

$$I(I;O) = H(I) + H(O) - H(I,O) \tag{7-4-3}$$

The word 'uncertainty' used throughout the explanation illustrates that none of the parameters are static and everything is probabilistic in information theory. Shannon's noiseless coding theorem (Shannon, 1948) states that it is possible to achieve errorless transmission if the information transfer rate is less than the channel capacity. Although the above statement is not completely true in some circumstances, it has been proven that errorless transmission can be achieved by adding some redundancy, as explained in Shannon's noisy coding theorem in the

same paper. Any designed method of error protection must maintain the information rate less than the channel capacity in line with Shannon's theorem.

The channel capacity is derived from the mutual information. The channel capacity can be described as the maximum mutual information that can be achieved over a channel with a particular transition matrix. The channel capacity is the maximum information transfer with high reliability that can be achieved across a particular channel. The channel capacity can be calculated by maximising the mutual information with respect to the source symbol probabilities. This implies that the channel capacity changes with the channel matrix and source probability.

In our research scenario, if the information theory concepts are approached from the channel coding layer, the channel matrix will represent the uncertainty of the source symbols. Each entry will indicate the probability of identifying a particular source symbol in a group as a different symbol; the channel matrix will be square with the number of rows and columns equal to the number of possible source symbols. The channel matrix may or may not be symmetric depending on the modulation scheme used. A channel can be considered to be symmetric when the rows of the channel transition matrix are permutations of each other.

If the same information theory concepts are approached from the source coding layer, in which every symbol is a video block, then the channel matrix can be modelled differently. The video block can be successfully decoded only if the data integrity of the block is not lost. So, as every symbol is a video block, the output can be in only one of two states, namely, either successful or unsuccessful. Since a video block is a long string of bits, it is impossible for the combination of the bits to resemble another video block due to error. This will make the channel 'binary erasure' in nature. The channel matrix will not be a square, but it will be symmetric (7-4-3). Figure 7-6 shows the graphical representation of a binary erasure channel.

$$Channel\,matrix\_macroblock = P\left(\frac{O}{I}\right) = \begin{array}{c} \\ i_1 \\ i_2 \\ i_3 \\ - \\ i_n \end{array} \begin{array}{c} \begin{array}{cccccc} o_1 & o_2 & o_3 & & o_n & o_{n+1} \end{array} \\ \begin{bmatrix} P & 0 & 0 & -- & 0 & 1-P \\ 0 & P & 0 & -- & 0 & 1-P \\ 0 & 0 & P & -- & 0 & 1-P \\ - & - & - & -- & - & - \\ 0 & 0 & 0 & -- & P & 1-P \end{bmatrix} \end{array}$$

(7-4-3)

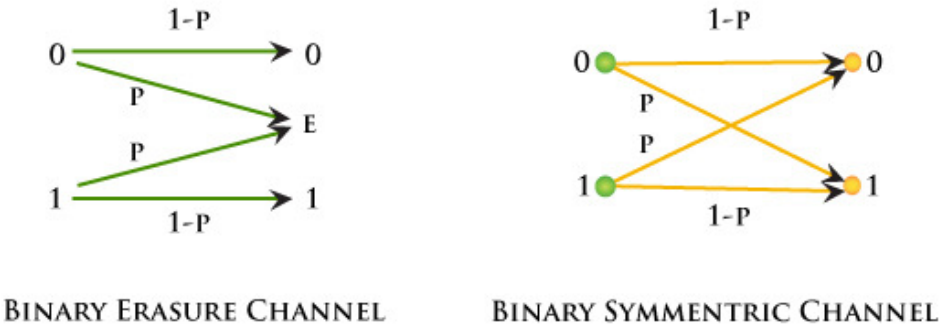


Figure 7-6. Binary erasure and symmetric channel

From the above equations, it can be observed that regardless of the channel error characteristics, the transition matrix is symmetric if a video block is considered to be a symbol. There is a well known result in information theory that the maximum information transfer can be achieved on a symmetric channel when the input symbols are uniformly distributed. This is because, when the input is uniformly distributed, then the output must be uniformly distributed. Though the channel matrix in both channel coding and source coding layers are symmetric, the uniform source distribution will have a different impact on both layers. If there is an error in a uniformly distributed source at the channel coding layer, it will result in current symbol being replaced by a symbol that belongs to the same code group, which is a valid codeword. It is not possible to predict if an error has occurred unless the data is decoded in the higher layer. Whereas, at source coding the error can be easily detected and rectified as it is impossible for a particular combination of bits representing a particular source symbol to be mistaken as another symbol. This is a very important observation, as it is used in the later part of the research.

The discussions presented in this section can be summarised as: -

- Errorless transmission is not possible if the information rate is more than the channel capacity.
- The transition matrix becomes symmetric when the basic unit is considered to be a macro-block.
- When the basic unit is a video block, the channel is binary erasure in nature rather than binary symmetric.
- When the channel matrix is symmetric, then the maximum information rate can be achieved when the source symbols are uniformly distributed.

7.5. A Basic Experiment to Compare Source and Channel Error Coding Methods

From the observations made in the previous section, an experimental setup was designed to test a simple source coding method against the channel codes.

Theoretically, the source coding and channel coding can be optimised separately to achieve the same performance (with some impractical assumptions) (Zhang et al., 2008).

The experimentation is aimed at proving that the above statement can be achieved practically.

The source coding method for the experimentation is designed as follows: -

- Since the basic unit, 'symbol', must be understandable by the application layer, a symbol is assumed to be a video block, which is the basic unit of a digital video coding system.
- As the channel matrix is symmetric from the higher layer's perspective, the distribution of the source is made uniform for maximum information transfer. This is accomplished by repeating the video blocks at uniform intervals in the video bitstream.

- Since the channel holds memory in higher layers, it is necessary that the coding system should hold some memory as well. Hence, the decoder knows that the source is uniformly distributed and also has knowledge of the error concealment process.

To make the comparison fair, the channel codes for performance comparison are chosen based on the code rate. A two-time repeated uniform source stream is compared to the convolutional code of code rate 'two'. Similarly, a three-time repeated uniform source stream is compared to the turbo code of code rate 'three', which is standardised for 3G WCDMA communications.

The decoding methodology used in the convolutional coding is Viterbi coding, whereas turbo codes use both log-likelihood and posteriori probability methods. The decoding method in the turbo coding utilises 'a priori' information along with the current observation. If the current observation is not strong enough to make a decision about a codeword, then the information from the previous decoding (priori) can be used. Unlike other coding schemes, turbo codes have the flexibility to predict the codeword from either current or priori information or both.

The decoding mechanism of the experimental source coding is based on the 'store and process' principle. If the decoder cannot decode a video block because of an error, it skips the video block and continues decoding other blocks. When the decoding point reaches the location of the repeated block, it patches up the missing block with the repeated block. Since the decoder knows the repetition rate of the blocks, as it holds some memory, if the error had affected all the repeated blocks, then it would use well known spatial and temporal concealment methods (Wang and Zhu, 1998) to patch the missing block. The spatial concealment method is used for intra-frames by stretching the neighbouring blocks and the temporal concealment is used for inter-frames by replacing the missing block with the corresponding block in the previous frame.

Detailed simulation results are presented in Chapter 9, some sample results are illustrated in Figures 7-7 and 7-8. The experimental results prove that the channel codes can be replaced by source coding methods for bursty mobile channels. The

results show that if data can be replicated uniformly ‘n’ times across the bitstream from the source coding layer, then it is possible to provide error resilience to a level offered by the channel codes of code rate ‘n’. It can be observed from figures that when the error characteristics are gradually changed from random to bursty, the performance of the source error methods over channel error methods gradually improves and outperforms the channel error methods when the errors are highly bursty in nature.

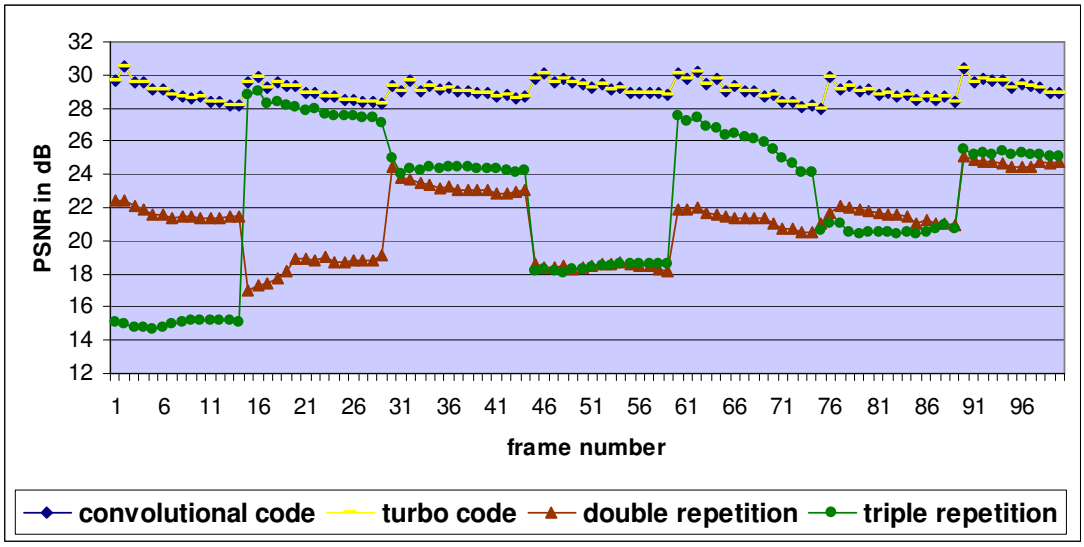


Figure 7-7. Effect of random errors on ‘foreman’ test clip

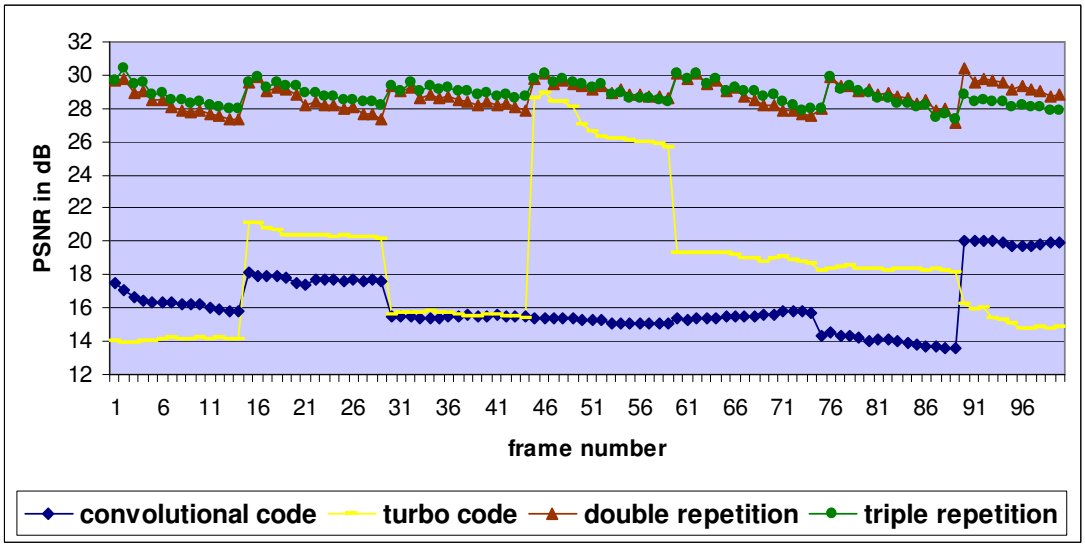


Figure 7-8. Effect of bursty errors on ‘foreman’ test clip

The core principle behind data replication is to make extra copies of the blocks available to the receiver. The number of times the data are replicated is directly proportional to increase in the bitrate; if the data are repeated twice, then the bitrate is doubled, whereas if the data are repeated thrice, then the bitrate is tripled. In some situations, twice repeated source coding outperformed the turbo codes; in other situations, a three-times repetition was required to outperform the turbo codes. The unpredictable nature of the channels makes the maximum limit on the amount of redundancy addition for errorless transmission unpredictable. From Shannon's theorem, it could be understood that for error-free transmission, the amount of redundancy must increase indefinitely, but this would bring the information rate to zero, so there must be an upper bound to the amount of redundancy that could be added. The methods must be designed in a way that maximum error resilience is offered with minimum redundancy bits.

In the channel coding layer, the code rate can be reduced by using a puncturing mechanism, but it reduces the effectiveness of the underlying channel coding method. The reason behind the repetition in the source coding method is to assure the decoder that there are few other blocks in the bitstream that convey statistical information about the current block, which may be used in the event of data loss due to errors. Using the flexible tools available in the source coding domain it is possible to distribute statistical information among multiple video blocks in the bitstream without increasing the bitrate significantly, and without degrading the effectiveness of the underlying source coding method.

This section justified by a simple experimentation that the source error control methods can provide error resilience to a level offered by the channel codes for the same amount of redundancy. The superiority of source error control methods over the channel coding methods cannot be justified if the level of redundancy required by both methods to achieve a certain level of performance is the same. Novel methods for reducing the redundancy in the video bitstream at the source coding layer without degrading the performance are explained in the next section.

7.6. Source Redundancy Reduction Methods

There are two methods by which the redundancy reduction can be accomplished from the source coding layer without degrading the performance:-

- external redundancy reduction.
- internal redundancy reduction.

Some internal redundancy methods under development are explained in this section and an external redundancy method that is completely developed is explained in the next chapter. The internal methods modify the coding structure of the video compression system, for example, changing the methodology of the prediction by using multiple reference pictures. A popular internal redundancy reduction is 'data hiding'. Data hiding is the process of hiding the statistical information about a particular video block in another video block without physical repetition. There are a few works on data hiding proposed by other researchers (Park et al., 1994; Gallant and Kossentini, 2001; Lee et al., 2005). Some simple data hiding techniques (Elangovan et al., 2008; Elangovan et al., 2007a) are proposed in this thesis. Since the methods are at a primitive stage, only the basic principles are explained in this thesis. Rigorous testing must be carried out to establish the robustness of the algorithms; due to time constraints, further work on the data hiding methods will be carried out as future work.

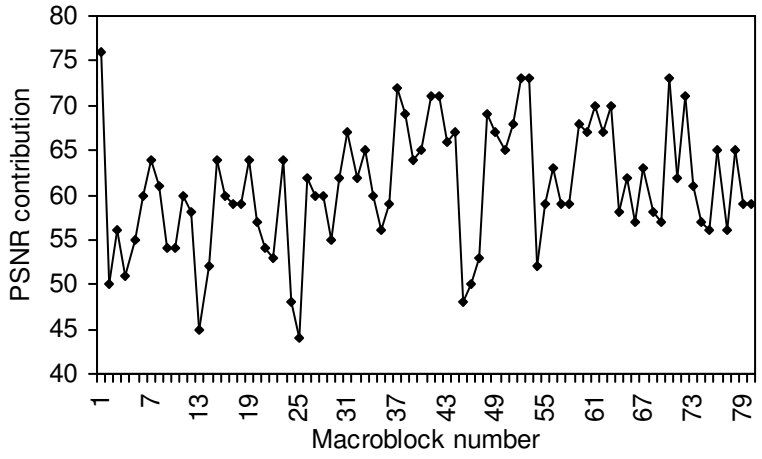


Figure 7-9. Quality contribution of the macro-blocks

The aim of the data hiding method proposed in (Elangovan et al., 2007a) is to equalise the quality contribution of the macro-blocks to the overall frame quality. The quality contribution each macro-block of the video frame is of a different level to the overall frame quality. A macro-block containing complicated texture information would demand more bits and contribute more to the overall frame quality, whereas a macro-block with uniform texture information would demand fewer bits and contribute less to the overall frame quality. The graph shown in Figure 7-9 illustates the quality contribution of different macro-blocks from frame 20 of the foreman test sequence.

The methodology is based on the principle of motion vector distribution and residual shuffling. The method aims to distribute the information, so that in the event of data loss due to errors, it would be possible to extract the missing information from a different section of the video bitstream to recover the lost data. After processing, each motion vector will have at least one pair, which can provide statistical information about each other. If motion vectors mv_i and mv_{i+x} are two motion vectors seperated by a distance x, they can be said to be natural pairs if they satisfy one of the following heuristic conditions (7-6-1).

$$mv_i = mv_{i+x}$$
$$mv_i = \overline{mv}_{i+x}$$
$$mv_i + mv_{i+x} = mv_D$$
$$mv_i - mv_{i+x} = mv_D \qquad (7\text{-}6\text{-}1)$$

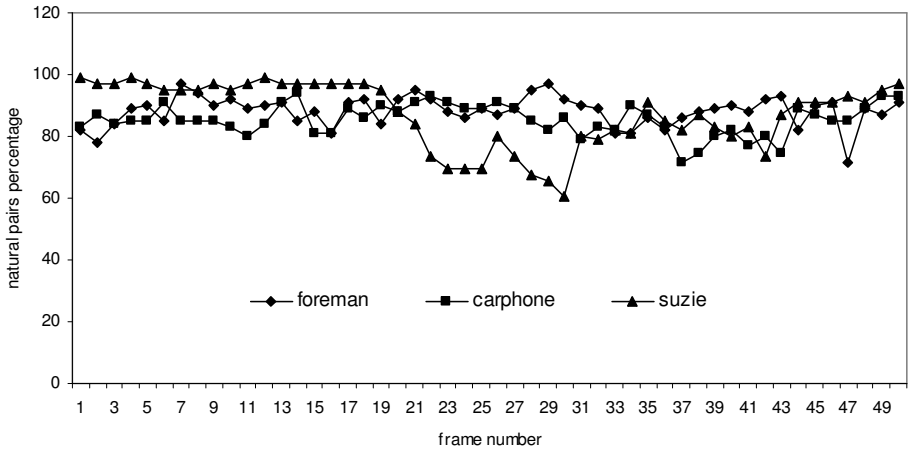


Figure 7-10. Percentage of natural pair occurrence

The dominant motion vector (mv_D) of a frame is defined as the vector that has the largest probablity of reoccurance. From repeated simulations on various test sequences, it was found that approximately 90% of the motion vectors occur as natural pairs satisfying the equations (7-6-1) by default. The graph shown in Figure 7-10 supports the argument. The motion vectors that do not satisfy the above conditions are optimised in order to satisfy conditions with a minimal increase in the residual information. The residual information is scrambled in a special pattern (horizontal, vertical or checkerboard) among the motion vector pairs.

The first four bits of the *mb_number* field are used to convey the control information about the scrambling and distribution modes. The macro-blocks are multiplexed with their pairs placed back to back in the bitstream, as shown in Figure 7-11. The length coding scheme used in the elementary stream headers of MPEG 4 is utilised for facilitating parallel decoding.

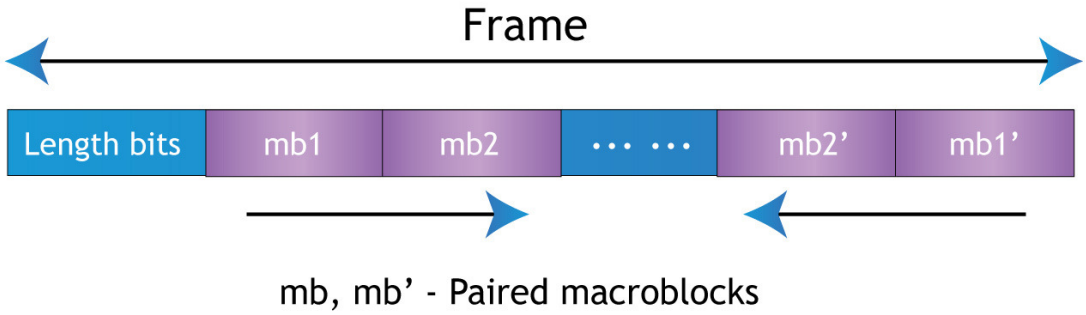


Figure 7-11. Back-to-back macro-block multiplexing

Though the algorithm proves its consistency in terms of quality and resource utilisation, the quality losses and the percentage increase in the bitrate challenge the overall performance. Although the primary objective of providing the error concealment in the source coding domain by data hiding is achieved with good subjective quality and with a less than 30% increase in the bitrate, additional effort must be made to justify its suitability across various video streams and channels of different characteristics.

The motion vector smoothing method proposed in (Elangovan et al., 2008), virtually repeats and interleaves the video data. The method is based on the principle of motion vector smoothing, where the motion vectors are processed on the basis of a data sharing principle and error propagation effects are neutralised by modified temporal prediction. The algorithm is based on some basic information theory principles: -

- If the probability of occurrence of a motion vector is less, then the information carried by it is more.
- For a frame containing 'n' motion vectors, the maximum error resilience is required if the motion vectors are equi-probable.

Smoothing of the motion vectors causes the neighbouring blocks to hold the same vector value. The motion prediction is carried on from the raw motion compensated previous frame after excluding the residual. A super-block is defined as a combination of four macro-blocks with a total pixel size of 32 x 32. If the probability of occurrence of a particular motion vector is less than a threshold value within a super-block, then the motion vector is optimised to take one of the values of the vectors of higher probability. The above-mentioned principle is applied only if the probability of reoccurrence is greater than 0.5. If the condition is not satisfied, the principle of global motion estimation is applied. This process involves the application of a motion estimation algorithm for the whole super-block and finding the motion vector that reduces the residual intensity.

To truncate the error propagation, the current frame is predicted from the motion compensated reference frame (assuming the residual to be zero). Equation (7-6-2) is used for normal prediction, whereas equation (7-6-3) is used for modified temporal prediction. Hence, if the lost motion vector of the erroneous block is reconstructed with absolute precision, then there is no possibility of error propagation.

$$f_{reference} = f_{motion\ \ compensated} + f_{residual} \tag{7-6-2}$$

$$f_{reference} = f_{motion\ \ compensated} \tag{7-6-3}$$

At the decoding end, if there were a loss of information, the motion vectors can easily be predicted from other copies within the super-block. Since the residual information is extracted from the raw motion compensated previous frame, there would be no error propagation if the missing vectors were reconstructed accurately. The information has been virtually repeated four times with minimal increase in the bitrate. The system efficiency is represented by equation (7-6-4). If the blocks are interleaved in such a way that the displacement length is greater than the average burst length, maximum efficiency could be achieved.

$$Efficiency = \frac{repetition\ \ rate * displacement\ length}{average\ \ burst\ \ length} \qquad (7\text{-}6\text{-}4)$$

The motion smoothing algorithm shows good performance improvement in terms of quality and bitrate. The data are replicated in the bitstream using data hiding methods with a reasonable increase in the bitrate (approximately 70% increase in the bitrate). Since the residual intensity is increased due to modified temporal prediction and the inter-block compression methods are used without special coding for DC components, there is a fractional reduction in the PSNR values. Further work must be carried out to make the method more robust by an intense testing process, as many parameters were manually fed into the system.

This section explained some ‘internal redundancy reduction’ methods, in which the redundancy reduction was achieved using ‘data hiding’ methods. On the other hand, the ‘external redundancy reduction’ methods modify the bitstream structure without changing the coding structure, for example, changing the methodology of packetization. Unlike partial methods explained for internal redundancy reduction in this section, there was sufficient time for the complete development of an external redundancy method. The successful optimisation procedure of resync markers for reducing redundancy (Elangovan et al., 2007b; Elangovan et al., 2006) is explained in the next chapter.

7.7. Summary

In this chapter an investigation was put forward to challenge the suitability of channel codes for applications that are highly data dependent and channels where the errors occur in clusters. Shannon's theorem explains that it is possible to achieve errorless transmission if some redundancy is added, but does not specify how and where the redundancy should be added. The above argument is proven in this chapter with a basic experiment where a simple source error resilient method that operates in the higher layer is compared to channel codes operating in the lower layer. These results will help to achieve the broad objectives discussed in the third chapter, in which it was explained that the error concealment block must be confined to the source coding block for ease of practical implementation. Finally, the chapter concluded with discussion on a number of internal redundancy reduction methods under development, in which the data can be replicated in the bitstream by using the flexibility available in the source coding layer without a significant increase in the bitrate. The next chapter presents a complete external redundancy reduction method, which will justify the convenience and flexibility of providing error protection from the source coding layer.

8. Virtual Partitioning of Compressed Video Streams by Invisible Resync Markers

The previous chapter stressed the importance of source error resilience over channel error resilience and, subsequently, some internal redundancy reduction methods were proposed. Every layer of the protocol stack designed for any application will have a certain form of error protection mechanism. For example, data link layer of the TCP/IP protocol stack utilises retransmissions, whereas transport layer utilises sequence numbers. The error protection mechanisms are designed in a way that it works the best on the form the data takes in that specific layer. Modifications can be imposed on the methods for better performance by further research. Though error protection can be offered from various layers of the protocol stack, the error protection offered at the application layer will have a significant impact on the final quality of the data. There were numerous error protection methods proposed to operate in the source coding layer. This chapter investigates a well known source error resilient tool, the 'resync marker', standardised for MPEG video coding, which is a fundamental component of a video packet, and external redundancy reduction methods are proposed for optimisation in order to achieve better performance. The results of investigations demonstrate the convenience of operating at the source layer of the protocol stack.

8.1. Introduction

Resync markers are special sequences inserted in the bitstream to restore the bitstream integrity, in the event of errors. When errors corrupt the bitstream, the decoder skips the data until the next valid resync marker and then resumes the decoding process. The data between successive resync markers are known as a video packet. There has been substantial research addressing the issue of choosing the appropriate location and the frequency of insertion of the resync markers, as they increase the bitrate of the compressed video stream. Lack of an efficient underlying algorithm to control the insertion process will challenge the compression efficiency of the video stream, as the addition of redundancy is very costly in terms of

bandwidth for a low bitrate video transmission. Since resync markers are simply a fixed sequence of bits, they are vulnerable to errors as well. Just increasing the number of resync markers does not guarantee better error resilience. An effective solution for the problem is proposed by a novel algorithm, 'virtual partitioning', where the resync information is conveyed virtually without using any bits.

8.2. Review of Literature and Chapter Organisation

When MPEG-2 was standardised, mobile multimedia technology was a decade away. This resulted in the standard being designed for reliable wired transmission. There are a few error resilient tools integrated into the standard, but they are not good enough to handle heavy data losses. The most popular MPEG-2 error resilient tool used is 'slice partitioning'. The smallest meaningful segment in a video context is a 'macro-block', and a 'slice' is a collection of macro-blocks. The restrictions imposed on the structure of the slice in the MPEG-2 are as follows:-

- The slice must contain macro-blocks that belong to the same rows.
- First and last macro-blocks in the slice cannot be skipped.
- The slice structure may differ from frame to frame.

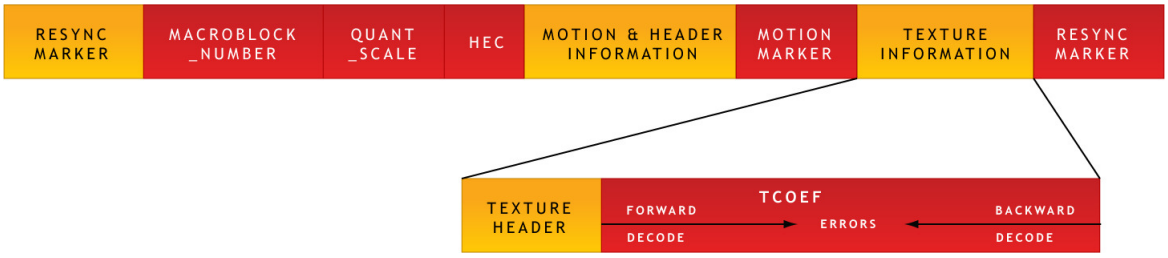


Figure 8-1. MPEG-4 video packet structure (ISO/IEC 14496-2 2001)

The MPEG-4 standard that followed MPEG-2 had sophisticated error resilient tools embedded in it. The MPEG-4 standard is widely used in the mobile multimedia transmission. The video codec used for low bit-rate streaming for third generation mobile phones has been extracted from the MPEG-4 standard. This section of the report gives a comprehensive review of the error resilience tools mentioned in Annex K of ISO/IEC 14496-2 MPEG 4 standard. The error resilient tools are listed below: -

- Resynchronisation Markers

- Data partitioning and Reversible VLC (RVLC)
- Adaptive Intra Refresh (AIR)
- NEWPRED mode

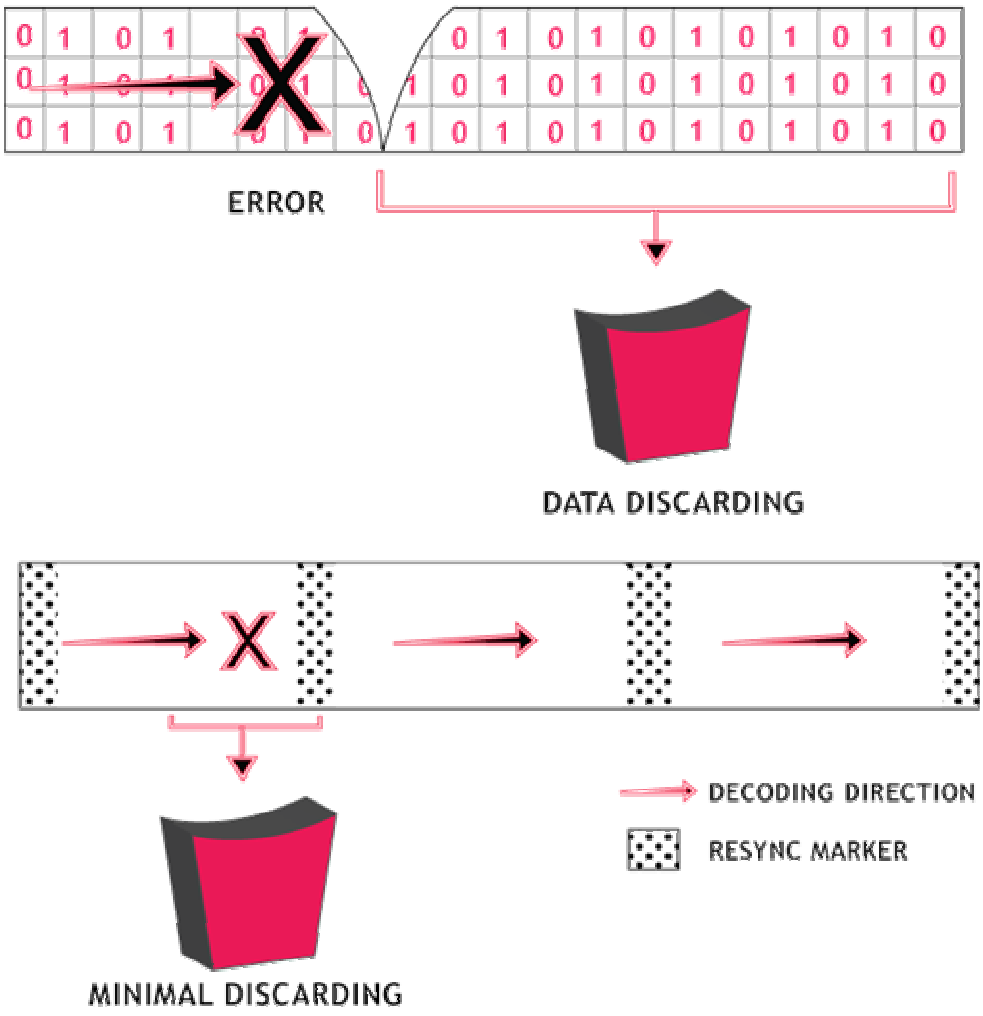


Figure 8-2. Resync marker operating principle

The structure of an MPEG-4 video packet is shown in Figure 8-1. The resync markers are periodically inserted in the bitstream to help the decoder to regain synchronisation. In general, the error is detected in the bitstream if an illegal VLC is received, more than 64 DCT coefficients are decoded in the block, the QP is out of range or if a semantic error is detected (ISO/IEC 14496-2, 2001). In the MPEG-4 standard, the data transmission is in the form of video packets and each packet starts with a *resync_marker*, which can be of 17-22 bits with a special pattern (a lengthy

string of zeros followed by a short string of ones). The marker is followed by the *macroblock_number* and the *quant_scale,* which convey the macro-block location and the quantisation information respectively. Extra control signalling information is embedded into the video packet to ensure the unique decodablity of the packet by *HEC* (Header Extension Code). In H.261 and H.263, the resync markers are inserted after every few macro-blocks, but this results in the resync marker positions being random due to the variable bitrate nature of the coding system. In MPEG-4, resync markers can be inserted uniformly across the bitstream at fixed intervals regardless of the macro-block position. This results in the resync markers being closer to each other in high activity regions and farther apart in the low activity areas (Fang and Chau, 2005). The resync marker's operating principle is graphically represented in Figure 8-2.

In addition to partitioning frame data into video packets using resync markers, the video packets can be partitioned further using MBM (Motion Boundary Marker). The data partitioning can be implemented by dividing the video packet into motion and texture information and inserting MBM between them. This will ensure that the errors occurring within a video packet are confined to one section of the packet (Yang et al., 2001). The texture information can be coded using RVLCs (Reversible Variable Length Codes). RVLCs can be decoded in both directions. If an error occurs in the bitstream, the decoder finds the next resync marker and starts decoding in the backward direction to minimise the information loss. However, RVLCs result in an increase in bitrate and decoding complexity (Moccagatta et al., 2000).

The AIR method involves the insertion of intra macro-blocks in an inter-frame. The logic behind the AIR method is to reduce the error propagation that results from high motion activity. The number of intra blocks in an inter frame can be predefined and, further, a static (Dogan et al., 2002) or dynamic memory map (Pang et al., 2004) may be used for consistency in the macro-block refreshing process. In NEWPRED mode, the reference image can be changed dynamically based on the feedback from the upstream channel (a logical backward channel opened between encoder and decoder).

However, with H.264, in spite of its compression efficiency, the error resilience offered by the standard is not as good as the MPEG-4. H.264 transmission can be synchronised by byte alignment of the Negative Abstraction Layer (NAL) units. The transmission can be made efficient by adding a start code and coding the size of the NAL unit using the *NumBytesInNALUnit* flag. H.264 has an on-board data partition module. The slice data can be partitioned into three segments and transmitted independently by signalling the *nal_unit_type* flag (2 - Partition A, 3 - Partition B, 4 - Partition C). The default partitioning scheme used in H.264 partitions the bitstream into three segments (A, B and C). The headers and motion vectors occupy partition A, intra coefficients occupy partition B and inter coefficients occupy partition C. The locations of the partitions are identified by *slice_id.* Some researchers have tried to improve the quality of transmission by applying UEP to the partitions (Barmada et al., 2005; Stockhammer and Bystrom, 2004) and by modifying the partitions (Dao and Fernando 2003). Other error resilient methods include redundant slice generation using multiple reference pictures and variable slice patterns, which are not suitable for low bitrate transmission.

Among all the error resilient tools previously explained, the resync marker has been proven to be practical due to its simplicity (Fang and Chau, 2005). Another significant advantage of the resync markers compared to other methods is that a change in the frequency of the insertion and the number of the resync markers in a bitstream will not require any change in the video coding syntax; this makes the tool very robust and more flexible than other methods (Gao and Tu, 2003b).

This section of the report reviews the existing literature relating to research on resync markers. Most of the literature focuses on the optimum placement of the resync marker for the best decoding quality, while other methods aim to reduce redundancy. The method, Partial Backward Decodable Bitstream (PBDBS) proposed by Gao and Tu (2003b) uses the simple logic of reversing the bits between two resync markers. If errors occur in the video stream and it becomes undecodable, then the decoder can browse through the bitstream to locate the next resync marker and then decoding can be resumed in the reverse direction. The graphical representation of the method is shown in Figure 8-3.

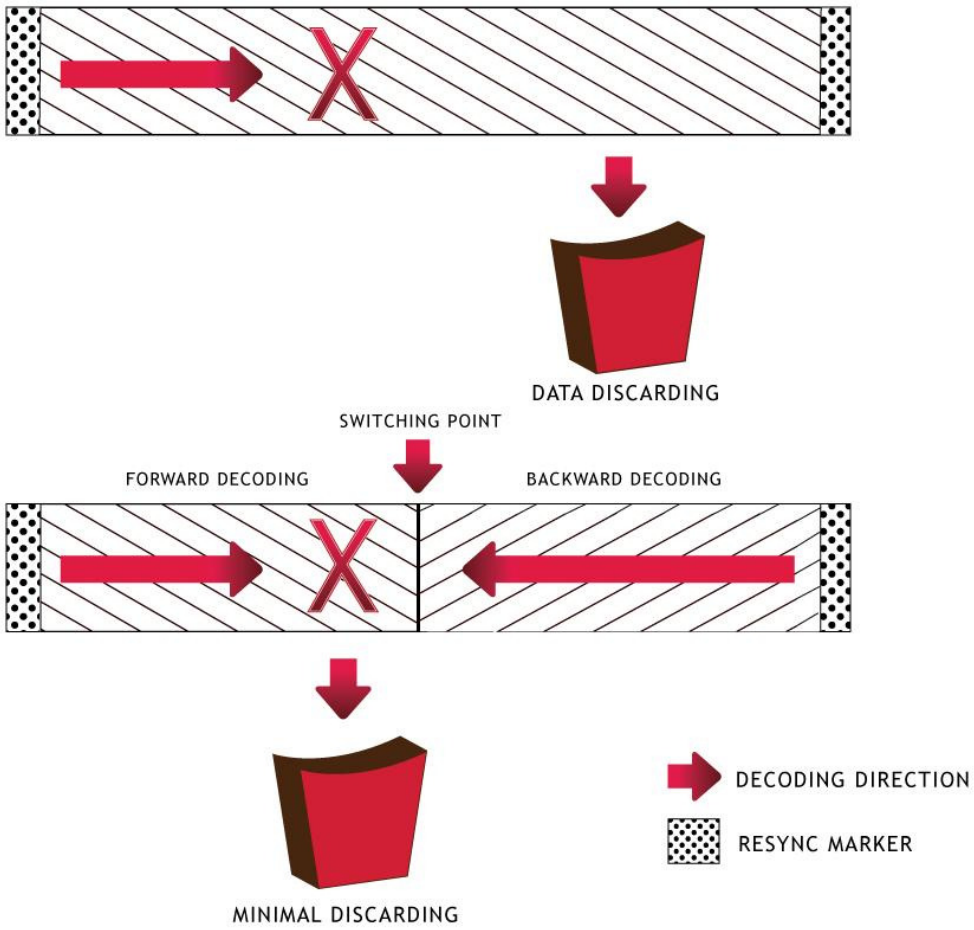


Figure 8-3. Partial Backward Decodable Bitstream (Gao and Tu 2003b)

The PBDBS method was modified by Fang and Chau (2005) to PRVLC (Partial Reversible Variable Length Codes). The primary difference between the two systems is the placement of the resync marker in the stream. In PRVLC, each slice is partitioned into foreground and background information using a threshold function. Then, the markers are placed between the foreground and the background data; this position also serves as the switching point for PDBMS. Gao and Tu (2003a) proposed another method to gain resynchronisation to the erroneous bitstream by estimating the bit count of the neighbouring macro-blocks. The point at which decoding should be resumed was identified by predicting the boundary of the next decodable macro-block from the bit counts of the previous macro-blocks. The method is efficient for intra macro-blocks, but not for inter macro-blocks. Since the exact location could not be identified in a single iteration, it takes many iterations to regain resynchronisation. Multiple loops will increase the decoder complexity and

time delay, which, in turn, will challenge the hardware efficiency. The method is graphically illustrated in Figure 8-4.

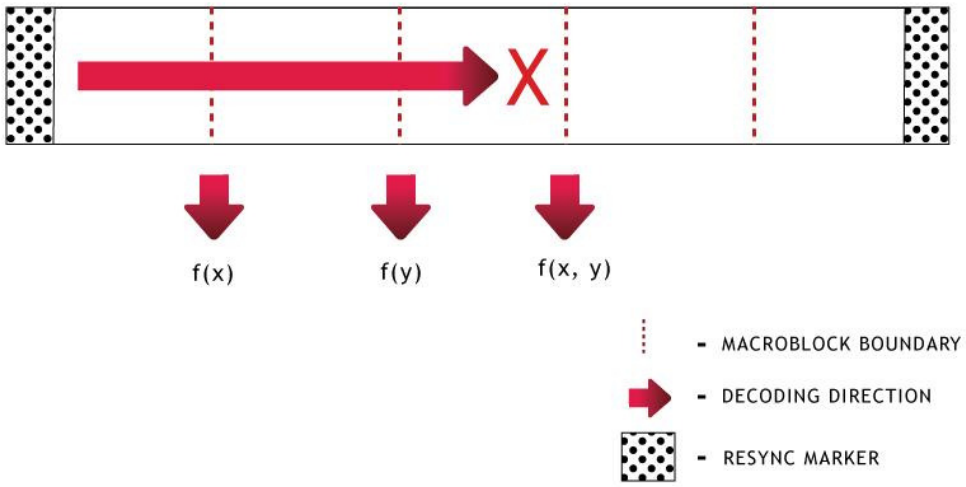


Figure 8-4. Boundary prediction method (Gao and Tu 2003a)

Yang et al. (2001) improved the overall quality of the wireless video transmission by optimising parameters like resync marker placement, intra refreshment and the coding mode of the macro-blocks. The problem was defined using the Lagrangian method and solved using dynamic programming. A similar method was proposed by Lee et al. (2001) in which the optimisation was carried out between the number of bits and the distortion measure of the blocks. Figure 8-5 shows the graphical illustration of the optimisation methods.

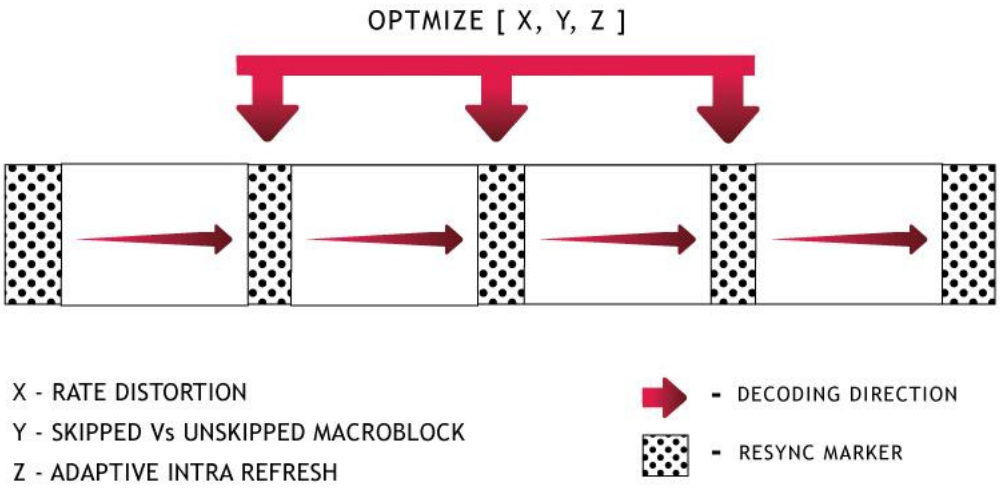


Figure 8-5. Resync optimisation methods

In section 8.3, the resync markers are analysed in detail. Subsequently, the relation between the number of resync markers in the video stream and the error resilience is investigated along with the problem definition. In section 8.4, the virtual partitioning method is proposed, which explains the process of facilitating the resync marker's functionality without a physical presence in the bitstream. The following section presents a modified rate-matching algorithm, which explains the process of accomplishing the virtual partitioning method without a significant increase in the bitrate. Section 8.6 concludes the chapter with a summary of all the investigations presented in the chapter.

8.3. The Resync Marker - An In-Depth Analysis and Problem Definition

This section investigates the resync marker in detail and challenges the argument that an increase in the number of resync markers will increase the error resilience. The above argument will theoretically hold true, but practically, an increase in the number of resync markers has a major downside in addition to the increase in redundancy. Let us consider a video packet containing 'vp_{mb}' macro-blocks preceded by a resync marker. A successful decoding of a macro-block within a video packet depends on the successful decoding of the previous macro-blocks, because of the variable length coding. The impact of errors on a video packet is explained with some mathematical equations. Taking the total size of the video packet to be 'vp_{length}', the video packet will satisfy the equation (8-3-1).

$$mb_{size} \neq \frac{vp_{length}}{vp_{mb}} \qquad (8\text{-}3\text{-}1)$$

The mb_{size} is the size of one macro-block. Let us consider two macro-blocks next to each other, and then the probability theory concepts can be used to explain the decoding process. There are four possible scenarios: -

- Scenario 1: Successful decoding of mb_n and successful decoding of mb_{n+1}

- Scenario 2: Unsuccessful decoding of mb_n and unsuccessful decoding of mb_{n+1}
- Scenario 3: Successful decoding of mb_n and unsuccessful decoding of mb_{n+1}
- Scenario 4: Unsuccessful decoding of mb_n and successful decoding of mb_{n+1}

Scenario 1 occurs when there is no error in the stream and both macro-blocks are successfully decoded. Scenario 2 is more likely to occur when there is an error in the mb_n and this results in mb_{n+1} being undecodable due to the variable length coding principle. It can be seen that the two events mb_n and mb_{n+1} are not mutually independent. Using probability theorems, the conditional probability of decoding mb_{n+1} given mb_n has already been decoded, as given in equation (8-3-2).

$$P\left(\frac{mb_{n+1}}{mb_n}\right)=\frac{P\left(mb_n \cap mb_{n+1}\right)}{P\left(mb_n\right)} \tag{8-3-2}$$

Scenario 3 results when the error occurs in mb_{n+1}; this will not have any influence on the decodablity of the mb_n, so these two events are mutually independent. The conditional probability of mb_n being decoded when mb_{n+1} has an error is given by equation (8-3-3).

$$P\left(\frac{mb_n}{mb_{n+1}}\right)=\frac{P\left(mb_n \cap mb_{n+1}\right)}{P\left(mb_{n+1}\right)} \tag{8-3-3}$$

If two events are mutually independent then equation (8-3-4) holds true.

$$P\left(mb_n \cap mb_{n+1}\right)=P\left(mb_n\right)\ P\left(mb_{n+1}\right) \tag{8-3-4}$$

Substituting equation (8-3-4) in (8-3-3) gives:

$$P\left(\frac{mb_n}{mb_{n+1}}\right)=\frac{P\left(mb_n\right)\ P\left(mb_{n+1}\right)}{P\left(mb_{n+1}\right)} \tag{8-3-5}$$

So, the probability of mb_n being decoded given that mb_{n+1} is already decoded is nothing but the probability of mb_n itself, as shown in equation (8-3-6).

$$P\left(\frac{mb_n}{mb_{n+1}}\right) = P(mb_n) \tag{8-3-6}$$

Scenario 4 is an impossible event; if the preceding macro-block mb_n has not been decoded successfully, it is very unlikely that the mb_{n+1} will be decoded successfully. So, the decoding process of the macro-blocks within the video packet will satisfy equation (8-3-7); the probability of successful decoding of a video packet is not equal to the product of the individual probabilities of the macro-blocks contained in the packet.

$$P(mb_1 \cap mb_2 \cap mb_3 \ldots\ldots\ldots \cap mb_n) \neq P(mb_1)\ P(mb_2)\ldots\ldots\ldots\ldots P(mb_n) \tag{8-3-7}$$

From the previous discussion, it can be understood that the maximum error resilience can be achieved if the video packet contains just one macro-block. The results from experimentation carried out by Belda et al. (2006) prove that in high error rate conditions, a smaller video packet results in a better performance than large video packets. However, a video packet of size one will challenge the compression efficiency of the system due to additional redundancy. There will be 50 kbps of redundant information to be transmitted for a QCIF (Quarter Common Intermediate Format) video stream with a frame rate of 15 frames/sec, if 22 bit resync markers and 12 bit macro-block number fields are inserted into the bitstream.

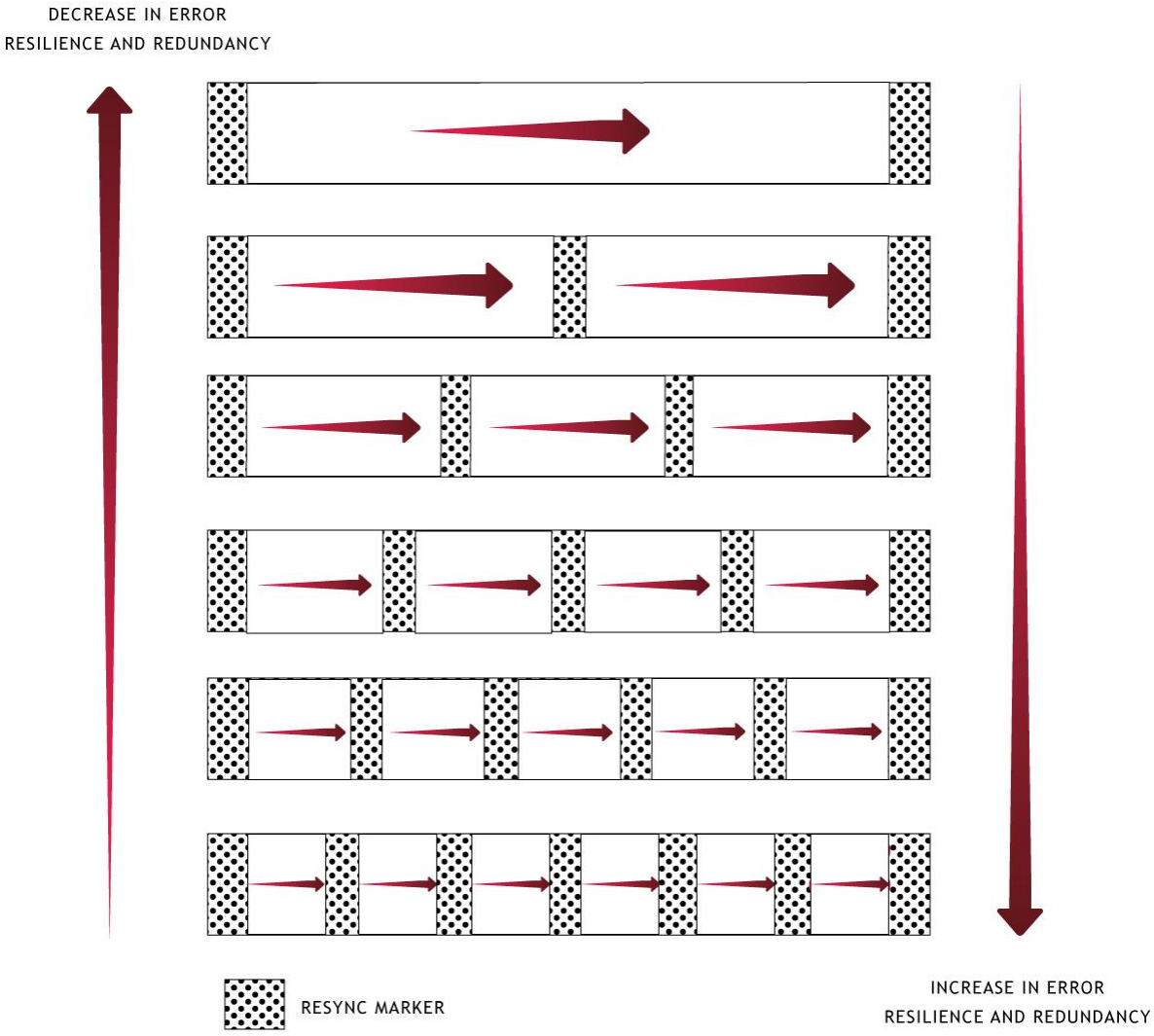


Figure 8-6. Number of markers vs error resilience vs redundancy

Figure 8-6 shows the relationship between the number of markers in the bitstream, resulting in error resilience and redundancy. Although this statement is misleading, most researchers agree with the above relation because of the theoretical soundness. The resync markers are simply a special sequence of bits, which has no special shield against errors. The bits in the resync marker have the same probability of error as have any other bits in the video packet. Increasing the number of resync markers will shift the problem in the opposite direction: the errors in the resync markers will affect the decodablity of a video packet that is not affected by the errors. This is because the decoder does not know the location of the resync marker

in the stream, so if it comes across an erroneous resync marker, it will treat it as a semantic error and skip the data until the next valid resync marker

The total number of bits generated on encoding a frame is the sum of bits generated by the resync markers and bits generated by the macro-blocks (8-3-8).

$$Frame_{bits} = R_{bits} + mb_{bits} \tag{8-3-8}$$

If 'x' is the number of resync markers and 'y' is the number of macro-blocks, then the total number of bits used for coding a frame can be represented by equation (8-3-11).

$$R_{bits} = \sum_{k=1}^{x} R_k \tag{8-3-9}$$

$$mb_{bits} = \sum_{n=1}^{y} mb_n \tag{8-3-10}$$

$$Frame_{bits} = \sum_{k=1}^{x} R_k + \sum_{n=1}^{y} mb_n \tag{8-3-11}$$

The number of video packets generated on encoding a video frame can be given by equation (8-3-12). This applies for both equally spaced and unequally spaced resync markers.

$$VP_{count} = \frac{\sum_{n=1}^{y} mb_n}{\sum_{k=1}^{x} R_k} \tag{8-3-12}$$

From the previous discussion, the trade-off between the number of resync markers and the size of a video packet can be summarised as shown in equation (8-3-13).

$$Error\ resilience = \begin{cases} x = 0 & nil \\ x < y & optimal \\ x = y & \max imum \end{cases} \quad (8\text{-}3\text{-}13)$$

Assuming the size of a resync marker to be the same as that of a macro-block, equation (8-3-11) can be rewritten as equation (8-3-15). In later sections, it is explained that the above assumption holds true in many circumstances.

$$R_k = mb_n \quad (8\text{-}3\text{-}14)$$

$$Frame_{bits} = \sum_{k=1}^{x+y} R_k = \sum_{n=1}^{x+y} mb_n \quad (8\text{-}3\text{-}15)$$

If the noise or error characteristics are unknown, then every block is equally likely to be affected by errors, and so the error probability of each macro-block or resync marker can be given by equation (8-3-16).

$$P_e = \frac{1}{x+y} \quad (8\text{-}3\text{-}16)$$

The probability of error in a frame is the summation of the probability of errors occurring in the resync markers and macro-blocks as shown in equations (8-3-17, 8-3-18).

$$P_{e\_frame} = P_{e\_resync} + P_{e\_mb} \quad (8\text{-}3\text{-}17)$$

$$P_{e\_frame} = (P_e * x) + (P_e * y) \quad (8\text{-}3\text{-}18)$$

If the best case resync marker method is implemented, where the size of the video packet is 1, then the number of resync markers is equal to the number of macro-blocks. The probability of error can be represented by equations (8-3-19, 8-3-20, 8-3-21, and 8-3-22).

$$P_e = \frac{1}{2x} = \frac{1}{2y} \quad (8\text{-}3\text{-}19)$$

$$P_e = P_{e\_resync} = P_{e\_mb} \quad (8\text{-}3\text{-}20)$$

$$P_{e\_frame} = 2*(P_e * x) = 2*(P_e * y) \quad (8\text{-}3\text{-}21)$$

$$P_{e\_frame} = 2*P_{e\_resync} = 2*P_{e\_mb} \quad (8\text{-}3\text{-}22)$$

It can be observed from the results that the probability of error of both resync markers and the macro-blocks are the same. This implies that the excessive presence of resync markers will contribute to the non-decodablity of a video packet as much as it contributes to its protection. The excessive presence of the resync markers increases the bit concentration of the marker bits and so does the probability of error. This establishes that if error protection is applied by something that demands physical presence in the bitstream, then the protection will not as effective.

It can also be observed from the discussion on existing methods from section 8.2 that they rely on the resync markers one way or another for decoding. This contradicts the discussion presented earlier in the chapter that if the error protection mechanism assumes something that is physically present in the bitstream to be immune from the errors, then that is not a valid assumption. The next section explains a novel method, in which the resync markers are replaced by invisible markers.

8.4. The Virtual Partitioning Method

This section explains the virtual partitioning method, where the resync markers are emulated without physical presence in the bitstream. The primary objective of using a resync marker is to re-synchronise with the bitstream in the event of errors, in other words, finding the boundary of the next decodable macro-block. Let us consider the macro-block boundary to be a random variable, then the probability of finding a macro-block boundary at random is uniformly distributed, due to variable length coding. This condition implies that the probability of each bit position being a boundary is the same. This condition is mathematically illustrated in equation (8-4-1); $vp_{start\_location}$ and $vp_{end\_location}$ are the locations of the first and last bits in the video packet respectively.

$$f(x) = \begin{cases} \dfrac{1}{vp_{end\_location} - vp_{start\_location}} & for\ vp_{start\_location} \le x \le vp_{end\_location} \\ 0 & for\ vp_{start\_location} < x \ \ or\ x > vp_{end\_location} \end{cases} \quad (8\text{-}4\text{-}1)$$

The best decoding case will be when all the macro-blocks are of equal size, so that their boundaries will be evenly spaced in the stream. Since the locations of the macro-block boundaries are known in advance, the probability of the random variable will not take any values other than 1. If this condition is achieved, then the bitstream can be synchronised with no additional overhead. This condition is mathematically illustrated in equation (8-4-2).

$$f(x) = \begin{cases} 1 & for\ vp_{start\_location} \le x \le vp_{end\_location} \\ 0 & for\ vp_{start\_location} < x \ \ or\ x > vp_{end\_location} \end{cases} \quad (8\text{-}4\text{-}2)$$

In reality, the above condition cannot be achieved, as the size of each block is a function of block size, texture distribution, motion vectors and header data. The virtual partitioning method optimises the stream, so that the macro-block boundaries follow a common multiplicative factor (8-4-3); 'M' is the multiplicative factor. It can be seen that the probability of finding the boundary bit of the next macro-block is much higher than the scenario presented in equation (8-4-1).

$$f(x) = \begin{cases} \dfrac{M}{vp_{end\_location} - vp_{start\_location}} & for\ vp_{start\_location} \le x \le vp_{end\_location} \\ 0 & for\ vp_{start\_location} < x \ \ or\ x > vp_{end\_location} \end{cases} \quad (8\text{-}4\text{-}3)$$

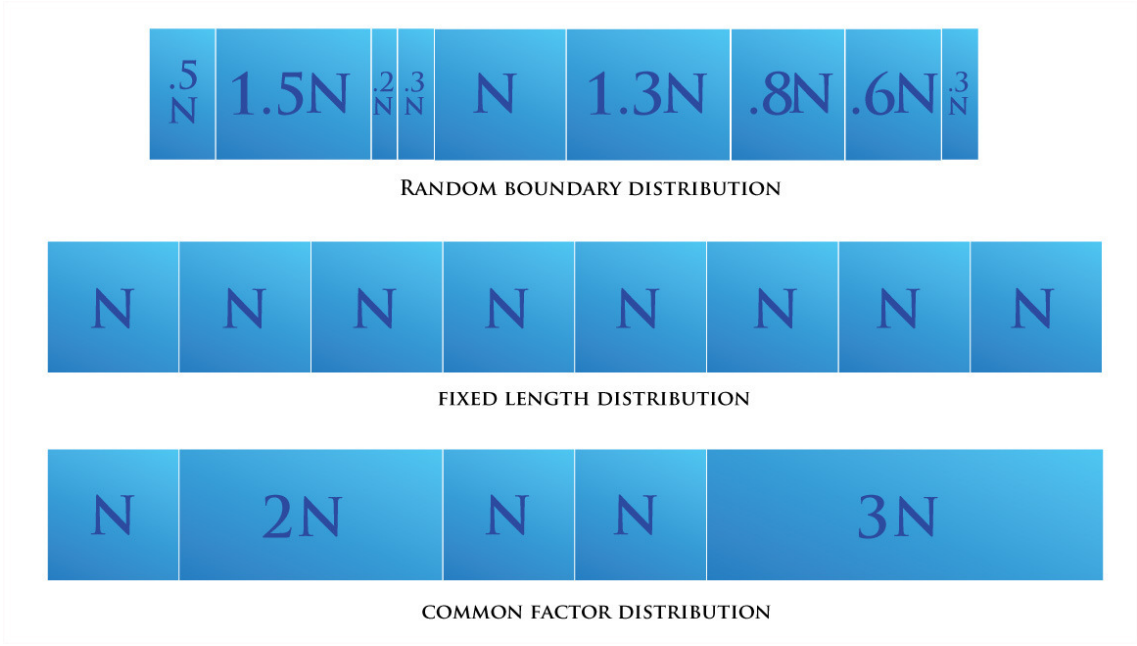


Figure 8-7. Modes of macro-block boundary distribution

This section explains the choice of an optimal multiplicative factor for a compressed video bitstream. Extending the mathematical model from the previous section, the number of bits for the resync markers remains the same regardless of the location of the markers. So the total number of bits for the resync markers can be represented by a constant as shown in equation (8-4-4).

$$R_{bits} = \sum_{k=1}^{x} R_k = C \tag{8-4-4}$$

Equation (8-3-15) can be rewritten as equation (8-4-5) by substituting equation (8-4-4).

$$Frame_{bits} = C + \sum_{n=1}^{y} mb_n \tag{8-4-5}$$

Let us consider ‘M’ as the common multiplicative factor that a macro-block boundary is required to follow. Since it is not an automatic process, the macro-blocks

are padded manually to follow the multiplicative rule. The padded macro-block that follows the multiplicative rule can be represented by equation (8-4-6); % is a 'modulo' operator. The number of padding bits added to a macro-block in order to satisfy the multiplicative rule can be represented by equation (8-4-7).

$$mb_{n\_padded} = mb_n + M - (mb_n \% M) \quad (8\text{-}4\text{-}6)$$

$$Padding_{bits} = \sum_{n=1}^{y} mb_{n\_padded} - \sum_{n=1}^{y} mb_n \quad (8\text{-}4\text{-}7)$$

Obviously, the process of padding bits will induce some redundancy; the maximum and minimum limit on the padding can be represented by the following equation (8-4-8).

$$Padding = \begin{cases} M - (mb_n \% M) = 0 & nil \\ M - (mb_n \% M) = M - 1 & \max imum \end{cases} \quad (8\text{-}4\text{-}8)$$

The above equation shows that no redundancy occurs when all the macro-block boundaries occur in the locations that satisfy the multiplicative rule by default. Maximum redundancy occurs if the all the macro-block boundaries occur at the first odd location following the location that satisfies the multiplicative rule. For optimality or to break even, the total number of padding bits added must be less than the redundancy imposed by the addition of resync markers. The following equations (8-4-9, 8-4-10) show that the break even occurs in redundancy when the multiplicative factor is less than or equal to the size of the resync marker.

$$M - (mb_n \% M) \le R_k \quad (8\text{-}4\text{-}9)$$

$$M \le R_k \quad (8\text{-}4\text{-}10)$$

$$R_k = \frac{R_{bits}}{VP_{count}} \quad (8\text{-}4\text{-}11)$$

The above conditions apply only if the size of the video packet is 1 or if it contains a single block (8-4-11). In very low bitrate video conditions, advanced rate matching algorithms are used and it may not be realistic to have a very short video

packet, in spite of poor channel conditions. In that case, the conditions shown in equation (8-4-12) will hold true and the multiplicative factor may differ. As the size of the video packet increases, the multiplicative factor decreases. The reduction in the multiplicative factor implies that there is an increase in the number of iterations before an exact macro-block boundary can be found.

$$R_k \neq \frac{R_{bits}}{VP_{count}} \tag{8-4-12}$$

This section explained a method of providing synchronisation without a physical presence in the bitstream. This method still adds some redundancy to the bitstream due to padding. The next section illustrates a novel method of accomplishing the above result without a significant increase in the bitrate, by modifying the rate-matching algorithm.

8.5. Virtual Partitioning by Modified Rate Matching

Rate matching is a process by which the number of bits allocated to the video frames is varied during compression in order to achieve a constant bitrate. This is accomplished by varying the quantisation parameter so that the amount of bits required for a coding a specific macro-block may be increased or reduced. The fundamental principle that drives the rate matching is the 'distortion measure'; this is a parameter that indicates the quality degradation that results due to variation in the QP. There are many methods in existence that accurately estimate the distortion measure of a macro-block. The investigation on rate matching algorithms in existence is beyond scope of the thesis. This section reviews the rate matching algorithm presented in the Annex L of the MPEG-4 standard codec. The rate matching process can be broadly classified into two: -

- frame rate matching
- macro-block rate matching

'Frame rate matching' allocates a specific QP to a frame, based on the number of bits available for compression; all the macro-blocks contained in the frame utilise the same QP. Equation (8-5-1) shows the generic method of allocation of bits to a

frame. It could be observed that the bit allocation process is a second order quadratic equation; X_1 and X_2 are model parameters.

$$Bits_{frame} = \frac{X_1\ MAD}{QP} + \frac{X_2\ MAD}{QP^2} \tag{8-5-1}$$

The 'macro-block rate matching' varies the QP on a macro-block-by-macro-block basis and the variation in the QP is signalled using the *DQUANT* parameter in the bitstream. Equation (8-5-2) shows the generic equation that represents the amount of bits generated by a macro-block; R is bits per pixel which is normally set to 0.085. Similarly, Equation (8-5-3) shows the amount of bits available to compress a specific macro-block; T_{Tex} is total bits available to code the texture information of a frame. Equation (8-5-3) shows that in very low bit-rate conditions, the QP remains constant for all the macro-blocks of a frame and in high bit-rate conditions, the QP increases with increase in MAD energy of the macro-block.

$$Bits_{mb} = \begin{cases} \frac{A_1}{QP_i^2} MAD_i^2 & Bitrate > R \\ \frac{A_2}{QP_i^2} MAD_i^2 + \frac{A_3}{QP_i} MAD_i^2 & Bitrate \le R \end{cases} \tag{8-5-2}$$

$$T_i = \frac{W_i\ MAD_i}{\sum_{j=1}^{N} W_j\ MAD_j} T_{Tex} \tag{8-5-3}$$

$$QP_i^2 = \begin{cases} MAD_i\ C_1 & Bitrate > R \\ C_2 & Bitrate \le R \end{cases} \tag{8-5-4}$$

The proposed method is independent of the rate matching algorithm used for the compression process. Once the QP for a specific macro-block is allocated by the rate matching algorithm the next step is to determine if the number of bits generated

by the macro-block for the allocated QP follows the multiplicative rule. If the multiplicative rule is not satisfied, the number of padding bits required for the macro-block to satisfy the multiplicative rule is determined. If the number of padding bits required is large, the QP is increased step-by-step until the number of bits required to compress the macro-block rolls back to the previous location that satisfies the multiplicative rule. For example, if the number of bits generated by a macro-block for a QP of 10 is 311 and the multiplicative factor is set to 20, the next bit location that satisfies the multiplicative rule in the forward direction is 320 and the previous bit location that satisfies the multiplicative rule in the backward direction is 300. In virtual partitioning method, explained in the previous section, the macro-block bits are padded in the forward direction to satisfy the multiplicative rule. In the modified rate matching algorithm, proposed in this section, the QP is varied so that the macro-block bits are rolled back in the backward direction to the previous point. The Pseudo code of the modified rate matching process is shown below:-

```
/*****PSEUDO CODE*****/
//Number of bits generated by a macro-bock for a specific QP
mb_bits = QP (macro_block)
//Number of padding bits required to satisfy the multiplicative rule
padding_bits  = multiplicative_factor -  modulo( mb_bits, multiplicative_factor)
// Next decoding point satisfying the multiplicative rule
next_point = mb_bits + padding_bits
//Previous decoding point satisfying the multiplicative rule
previous_point = mb_bits - modulo( mb_bits, multiplicative_factor)
// If the number of padding bits is greater then a threshold
if (padding_bits > threshold)
//Increase the QP until the macro-block bit count falls below the previous point
//satisfying multiplicative rule
while (mb_bits <= previous_point )
QP = Qp + 1
end
end
```

The increase in the QP will impact the reconstruction quality of the macro-blocks. Since the multiplicative factor is very small, one step increase in the QP is enough to roll the bits behind the previous point, in most cases. Although an increase in the QP will result in a fractional reduction in the PSNR, this reduction in quality is not significant as it is too small for human eye to capture the quality degradation at a high refresh rate. Essentially, the modified rate matching algorithm proposed in this section accomplishes the process of invisible marker insertion without any redundant bits in the bitstream by integrating the virtual partitioning method with the rate matching module during the compression process.

8.6. Summary

This chapter investigated in detail a source error control tool, the resync marker, and explained why addition of redundancy bits without an intelligent backbone algorithm can be inefficient. In the virtual partitioning method, the invisible markers are embedded into the bitstream to increase the robustness and to reduce the redundancy. Furthermore, the method is extended to exploit human perceptual principles so that the invisible markers can be inserted without any increase in the bitrate using a modified rate matching algorithm. The methods presented in this chapter are good examples of what could be accomplished from the source coding layer when the data are meaningful and understandable. The simulation results of the methods proposed in this chapter and the previous chapter are presented in the next chapter.

9. Simulation Results of the Algorithms Proposed to Solve Transmission Layer Issues

9.1. Introduction

This chapter presents the performance of the source error control methods compared to the channel error control methods proposed in Chapter 6 for channels of varying characteristics. This is followed by the results of the external redundancy reduction method proposed in Chapter 7 to reduce the redundancy imposed by the resync markers. In section 9.2, information on the test bench setup used for the simulations is presented along with information on the test clips. In section 9.3, the performances of the source and of the channel error control methods in random and bursty channels are presented. In section 9.4, the simulation results of the external redundancy reduction method, 'virtual partitioning', are presented along with the performance variation with various multiplicative factors used for the partitioning. In section 9.5, the results of the virtual partitioning method after integration of the 'modified rate matching' module are presented. Section 9.6 concludes the chapter with a summary of all the investigations presented in this chapter.

9.2. Matlab Test Bench

The video codec was programmed in Matlab, and the MPEG-4 simple profile codec described in the standard (ISO/IEC 14496-2, 2001) was chosen. The simple profile, as the name signifies, contains very basic features of the video codec. The features do not include advanced features like bi-directional frames, scalable coding and interlaced support. As it was a requirement of the research to understand every aspect of the video coding system to propose flawless error resilient methods that do not contradict the standard, the complete reference codec (C, C++ programs that come along with the standard) was rewritten in Matlab. This gave an understanding of the complicated codec and modification of the modules of interest, such as the 'resync marker insertion' and 'motion estimation' with greater flexibility.

Choosing a model to simulate the error characteristics of the mobile channels was a very difficult task. As explained in Chapter 2 much of the current literature claims a similarity to the real-time data. The MPEG-4 reference codec comes with a C-function code that simulates bursty losses using the Gilbert model. The code uses the 'rand' function to pick a random variable at every iteration; the total number of iterations corresponds to the total number of bits in the bitstream. If the random value is less than a particular threshold, then the corresponding bit is made erroneous; if the value is more than the threshold, then the corresponding bit is transmitted error-free. Many researchers have suggested that the optimum bit error ratio is approximately 10^{-3} for mobile wireless channels (Chou and Chen, 1996), indicating that one in every thousand bits is in error. In order to test the system against the worst case scenario, the bit error ratio is set to 10^{-2}, which represents a situation in which one in every hundred bits is erroneous. Since errors occur in clusters, the size of the cluster must be determined; the size of the error cluster depends on the carrier frequency, the Doppler frequency, the speed of the mobile and the transmission bitrate. From the link layer formulae it could be estimated that the average duration of the fade for a Doppler frequency of 40 Hz (vehicle travelling at 30 mph) is 1 msec. Assuming a transmission bitrate of 300 kbits/sec, the duration of the fade is 100 bits. The 'rand' function is uniformly distributed. To simulate the random errors, the threshold value is set to .001, and for simulating bursty errors, the threshold value is set to .0001.

The screen shots of the test clips used in the simulations are illustrated in Figure 9-1. The 'foreman' video sequence (9-1a), being a popular video test sequence, exhibits high motion due to camera shaking effects. The picture has a very high spatial detail. The 'carphone' video sequence (9-1b) has distinct foreground and background objects; the foreground object exhibits a random motion and the background object exhibits linear motion. This clip has a very high spatial detail, similar to the 'foreman' sequence. 'Suzie' (9-1c) and 'salesman' (9-1d) video clips are typical 'head and shoulder' video sequences. The background is static and the motion occurs with the foreground objects.

(a) Foreman clip

(b) Carphone clip

(c) Suzie clip

(d) Salesman clip

Figure 9-1. Test clips used in the simulations

The rate matching algorithm is used to control the bitrate of the compressed video streams. The video sequences are compressed to an average bitrate of 128 kbps. The average PSNR of the video sequences after compression is around 30dB. The quantisation parameter for the intra-frames is set to 15. The quantisation parameter for the inter-frames is varied dynamically by the rate matching algorithm according to the bitrate. The average quantisation parameter for the 'foreman' and 'carphone' video sequences was '12'; the average quantisation parameter for the 'Suzie' and 'salesman' video sequences was '8'.

The turbo codes used in the simulations employ the generator polynomials, [1,1,1; 1,0,1] with the code rate being 1/3. The data are processed in blocks of 500 bits and interleaved using the 'pseudo-random' interleaver. Similarly, the convolution codes employ the generator polynomials [1,1,0; 1,1,1], with the code

rate of ½. The data is processed in blocks of 500 and interleaved using the 'permutation' interleaver.

9.3. Comparison of Error Protection in the Channel and Source Coding Layers

This section presents the simulation results of the methods that provide error protection by the addition of redundancy from the channel coding layer and the source coding layers. The error protection in the channel coding layer is offered using turbo codes and convolutional codes. The error protection in the source coding layer is offered using repetitive coding. The channel is initially modelled with random errors, and subsequently, the channel characteristics are gradually varied by clustering the random errors to model bursty characteristics.

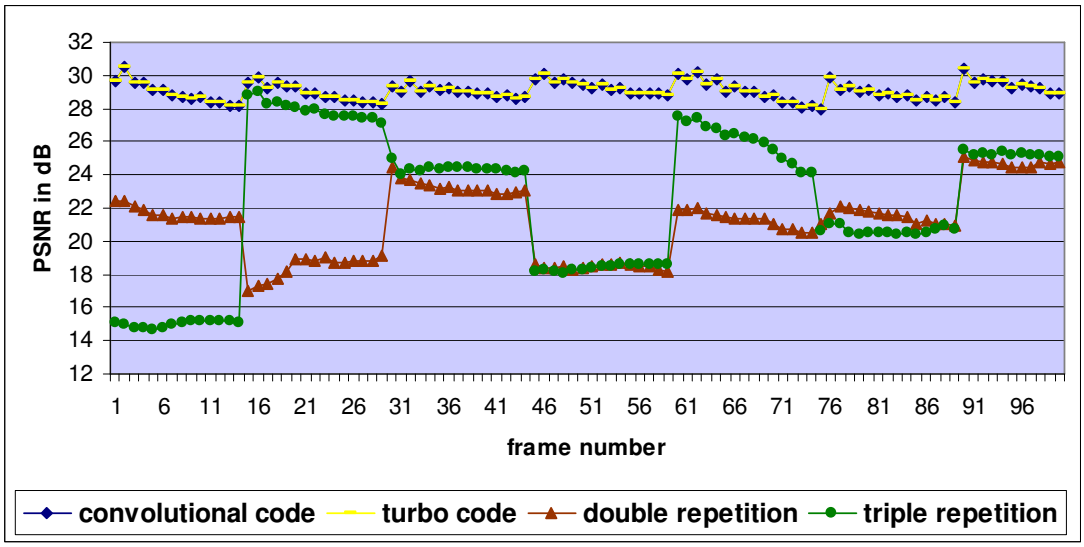


Figure 9-2. Effect of random errors on 'foreman' test clip

Figures 9-2, 9-3, 9-4 and 9-5 show the influence of random errors on the 'foreman', 'carphone', 'Suzie' and 'salesman' test clips respectively. The turbo codes and convolutional codes outperform the repetitive source coding methods. This is consistent across all the clips. The channel codes that are primarily designed for counteracting random errors are very efficient when the channel is characterised by one-bit errors. In bursty mobile channels, the errors are clustered, so the bits must be interleaved for channel codes to perform well. The length and type of the interleaver

plays a major role in the channel code's performance, because the interleaver must disintegrate the bursty errors and distribute them across the bitstream in a random fashion.

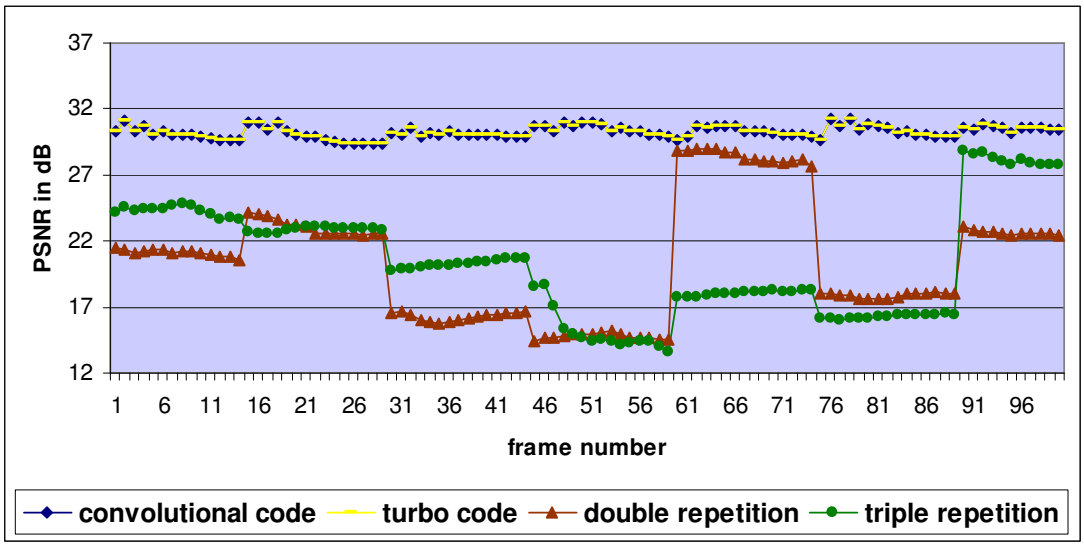


Figure 9-3. Effect of random errors on 'carphone' test clip

The repetitive coding is not efficient with random channels, as both the source blocks and the channel errors are uniformly distributed. This increases the probability of all the repetitive blocks corresponding to a specific area in a frame being erroneous.

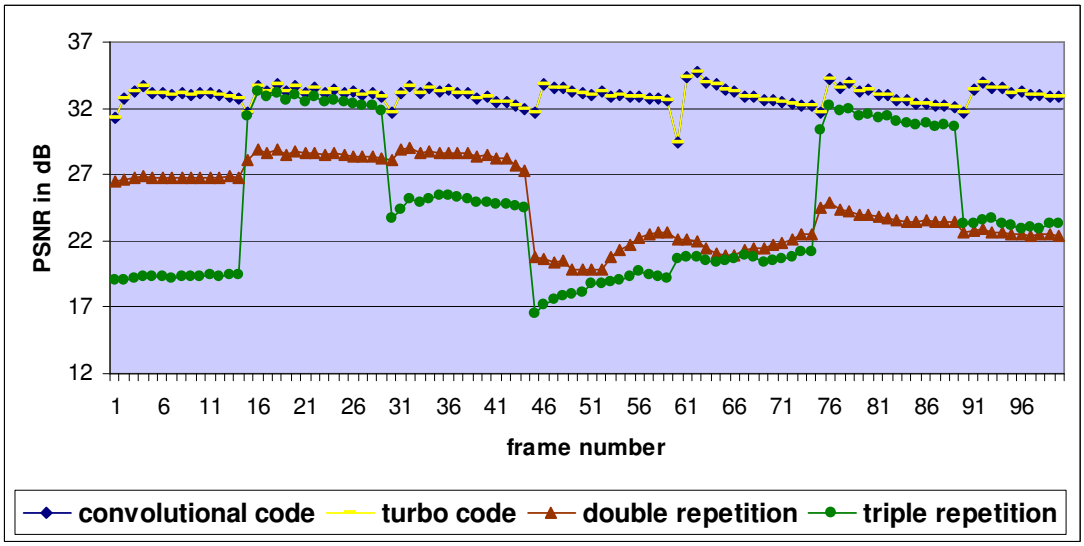


Figure 9-4. Effect of random errors on 'Suzie' test clip

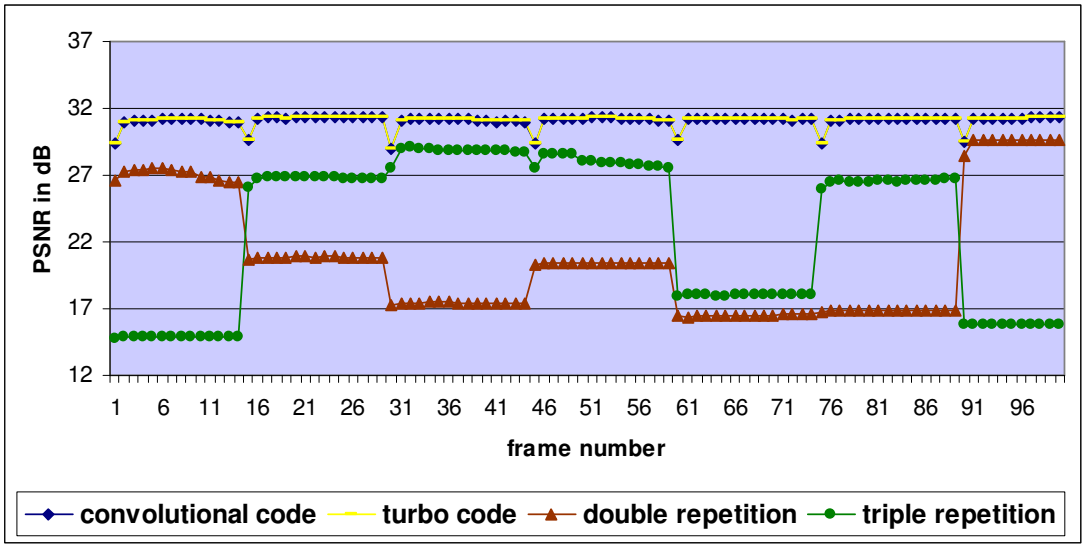


Figure 9-5. Effect of random errors on 'salesman' test clip

Figures 9-6, 9-7, 9-8 and 9-9 show the influence of bursty errors on the 'foreman', 'carphone', 'Suzie' and 'salesman' test clips respectively. It can be observed from the graphs that the repetitive coding outperforms the channel coding methods. In most cases, a two-time repetitive coding is enough to achieve the maximum performance. The channel codes degrade in performance with the bursty channels because of the inability of the interleaver to disintegrate the bursty errors into random errors. With repetitive coding, the channel errors are not uniformly distributed as the bursty errors are handled as they are, without interleaving, and the source blocks are uniformly distributed. This reduces the probability of all the repetitive blocks corresponding to a specific area in a frame being erroneous.

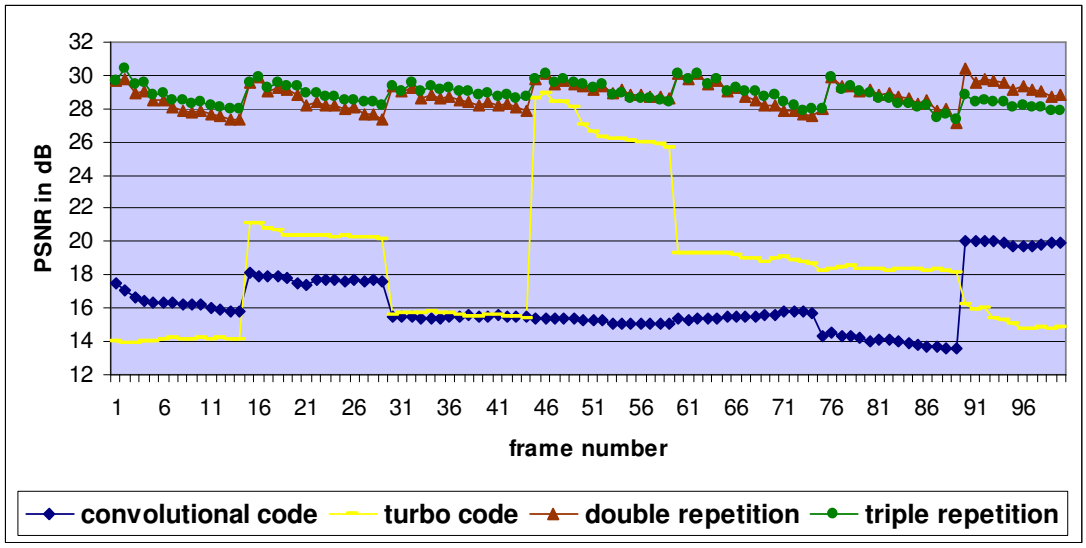


Figure 9-6. Effect of bursty errors on 'foreman' test clip

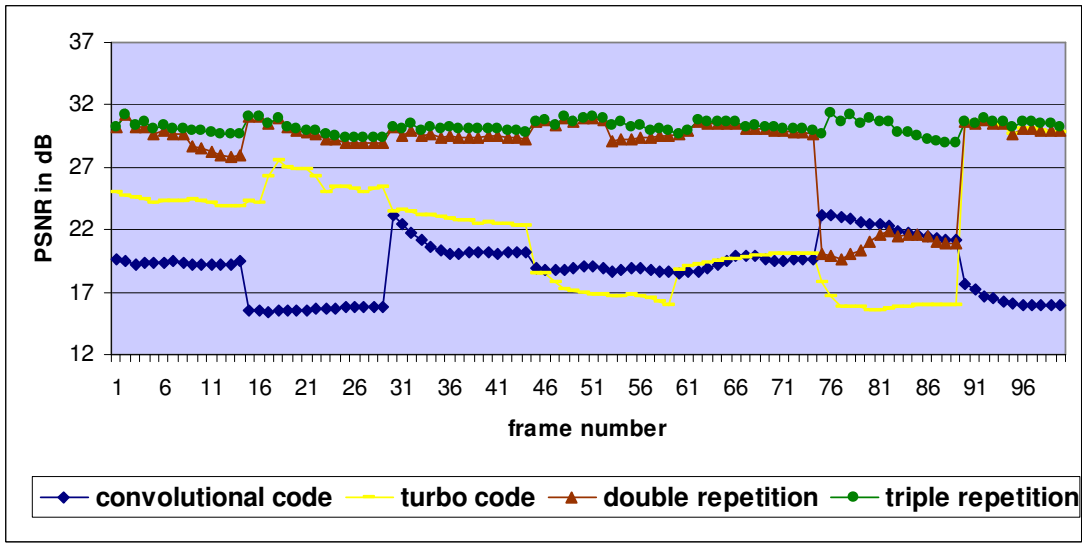


Figure 9-7. Effect of bursty errors on 'carphone' test clip

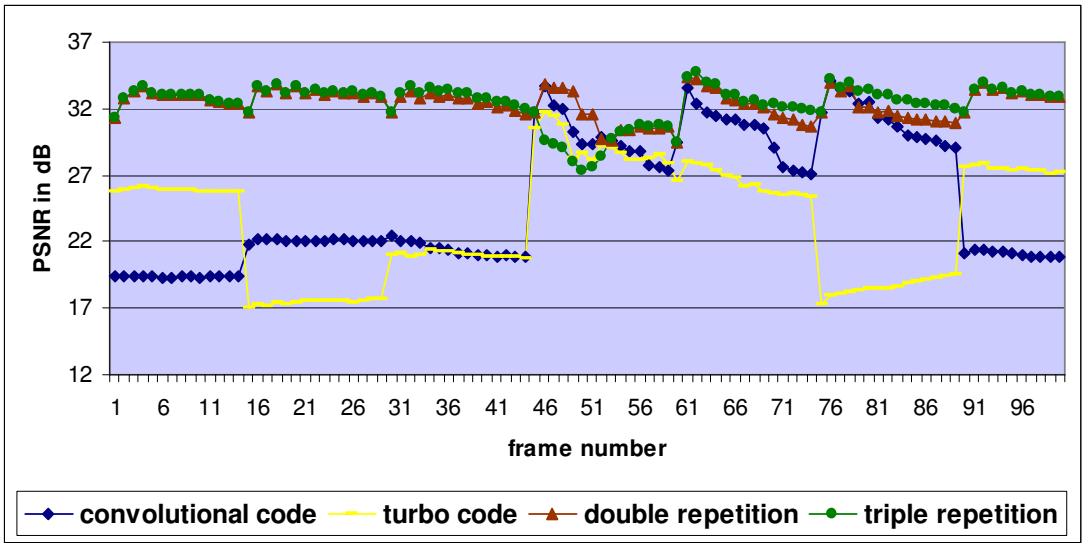


Figure 9-8. Effect of bursty errors on 'Suzie' test clip

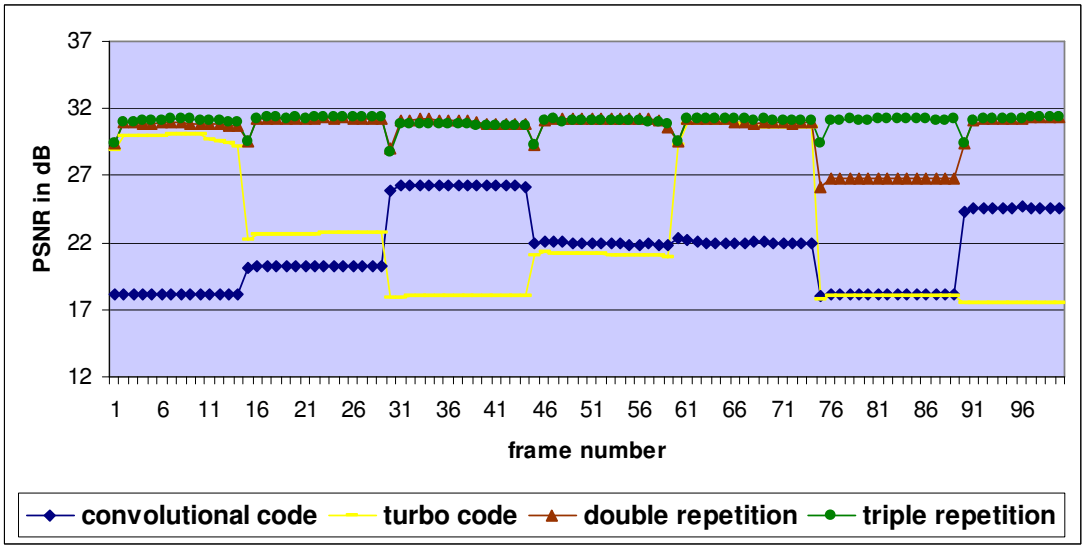


Figure 9-9. Effect of bursty errors on 'salesman' test clip

This section of the report explained how the error propagation in the mobile multimedia system can be effectively contained in the source coding layer rather than in the channel coding layer. As the error characteristics of the channel vary from random to bursty, the performance of the channel codes and source codes interchange in their respective performances. The performance analysis also proves that it is possible to achieve maximum performance with minimum redundancy in the source coding layer.

9.4. Simulation Results for Virtual Partitioning Method

This section explains the simulation results of the external redundancy reduction method, 'virtual partitioning'. The virtual partitioning method reduces the amount of redundancy imposed by the resync marker on the bitstream. Additionally, it emulates the functionality of the resync markers without the physical presence of any bits in the bitstream.

The proposed method is compared with three state-of-the-art methods, as listed below: -

- macro-block partitioning
- slice partitioning
- PBDBS (Partial Backward Decodable Bitstream)

In the 'macro-block partitioning' method, the size of the video packet is 1; in other words, the resync markers are inserted after every single macro-block. In the 'slice partitioning' method, the resync markers are inserted after coding a series of macro-blocks, preferably a row of macro-blocks. In PBDMS, the resync marker insertion process is same as in the 'slice partitioning' method, except the bits are reversed halfway though the video packet.

If errors cannot be imposed in the physical layer, this will result in inconsistent results because, since each method utilises a different methodology for resync marker insertion, the location of the macro-block in the bitstream differs across various methods. Accurate performance analysis can be performed only if the errors are imposed in the same location across bitstreams generated by different methods. This can be accomplished by imposing errors from the application layer of the protocol stack, so that the results produced will be consistent and accurate. A few erroneous macro-block locations are chosen with careful consideration given to their distribution (start, end and middle of the video packet).

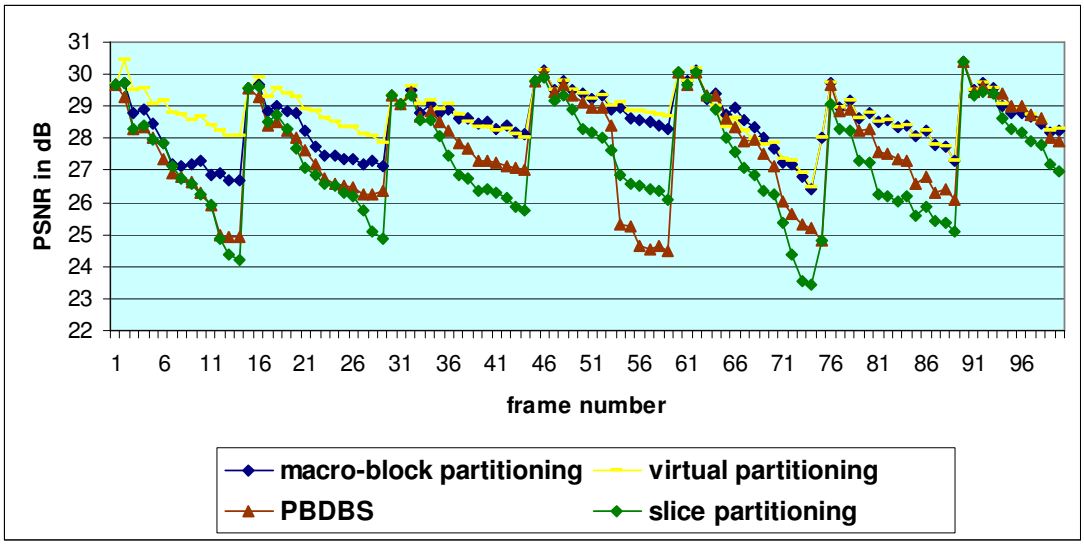


Figure 9-10: Comparison of existing resync marker methods and virtual partitioning on 'foreman' test clip

Figures 9-10, 9-11, 9-12 and 9-13 show the simulation results of various resync marker methods for different test clips. In general, it is observed from the graphs that the proposed method outperforms the existing methods. The macro-block partitioning method should theoretically result in maximum error resilience, as the video packet size is one. It can be observed from Figures 9-10 and 9-12 that the proposed method outperforms the macro-block partitioning method. This is because of the physical presence of the resync bits in the macro-block partitioning method. This highlights the advantage of the virtual partitioning method, in which no physical bits are used.

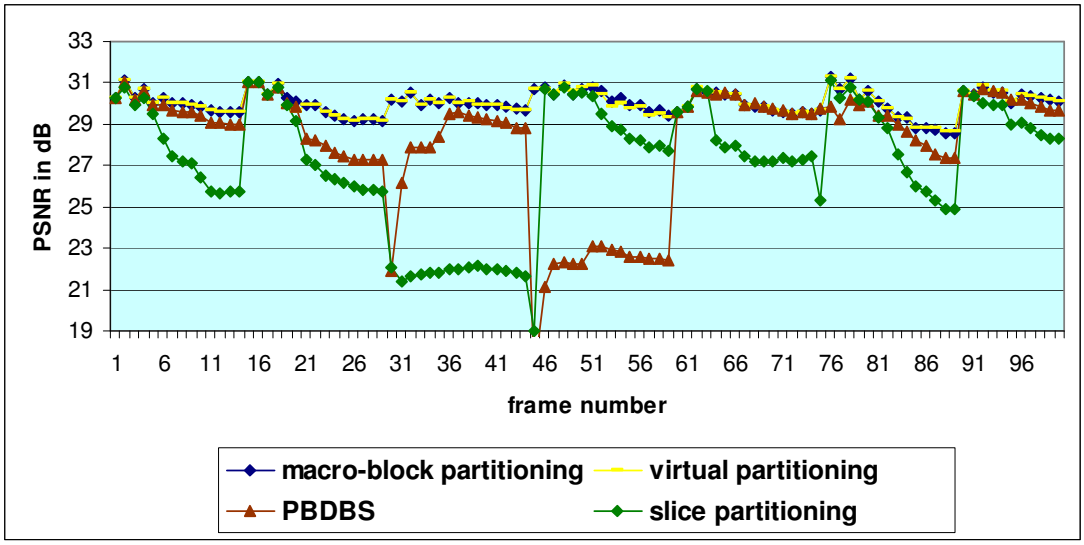


Figure 9-11. Comparison of existing resync marker methods and virtual partitioning on 'carphone' test clip

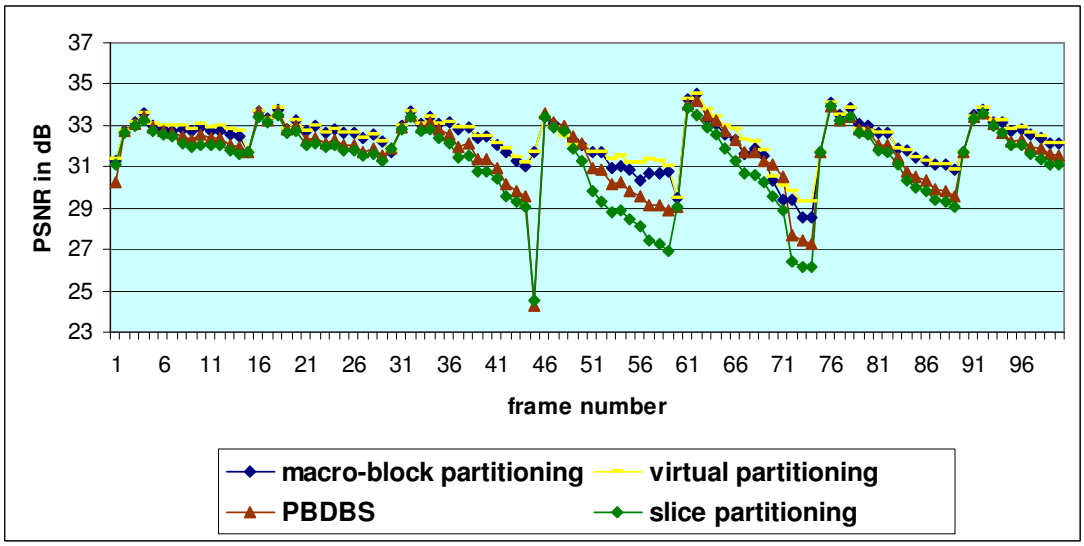


Figure 9-12. Comparison of existing resync marker methods and virtual partitioning on 'Suzie' test clip

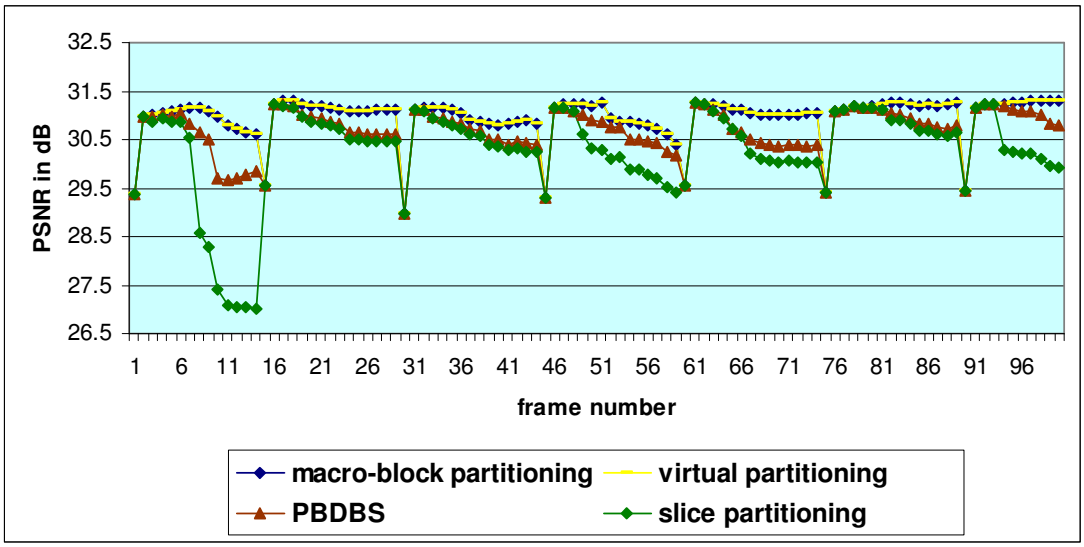


Figure 9-13. Comparison of existing resync marker methods and virtual partitioning on 'salesman' test clip

The PBDBS method outperforms the slice partitioning method, despite using the same amount of redundancy for resync markers. The GOP size is set to 15 while compressing the videos; this implies that an intra frame is inserted once every 15 frames. It can be observed in the figures that the curve peaks in once in every 15 frames due to intra-frame insertion and gradually decreases due to propagation errors. The error distribution characteristics explained in section 7.3.2 can be observed from the graphs. The overall characteristic of the error propagation is 'Exponential', but Figure 9-11 shows a 'Gaussian' distribution. The Gaussian distribution can be observed between frames 30 and 45 in the PBDBS method. An error in the intra-frame results in a sudden drop in the quality of the video, which is then shown to recover gradually from the loss. The Binomial distribution can be observed in Figure 9-12; an error results in a sudden drop in the quality of the video around frame number 73, and the quality instantaneously recovers as the next frame is intra coded.

Figures 9-14, 9-15, 9-16 and 9-17 show the amount of redundancy imposed by the resync markers on the video bitstream. The macro-block partitioning consumes the maximum number of bits, as the number of resync markers is equal to the

number of macro-blocks. The slice and PBDBS consume the same number of bits, as the number of resync markers inserted in both methods is the same. The amount of redundancy imposed by the proposed virtual partitioning method with a multiplicative factor of 20 is somewhere between the redundancy imposed by the macro-block partitioning and the redundancy imposed by the slice partitioning.

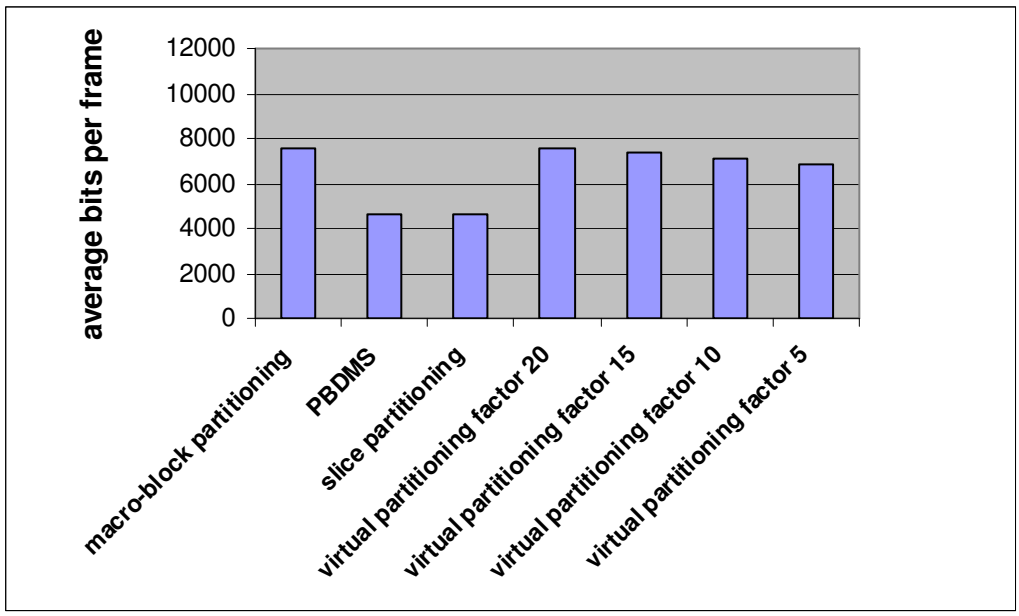


Figure 9-14. Redundancy imposed by existing methods and virtual partitioning on 'foreman' test clip

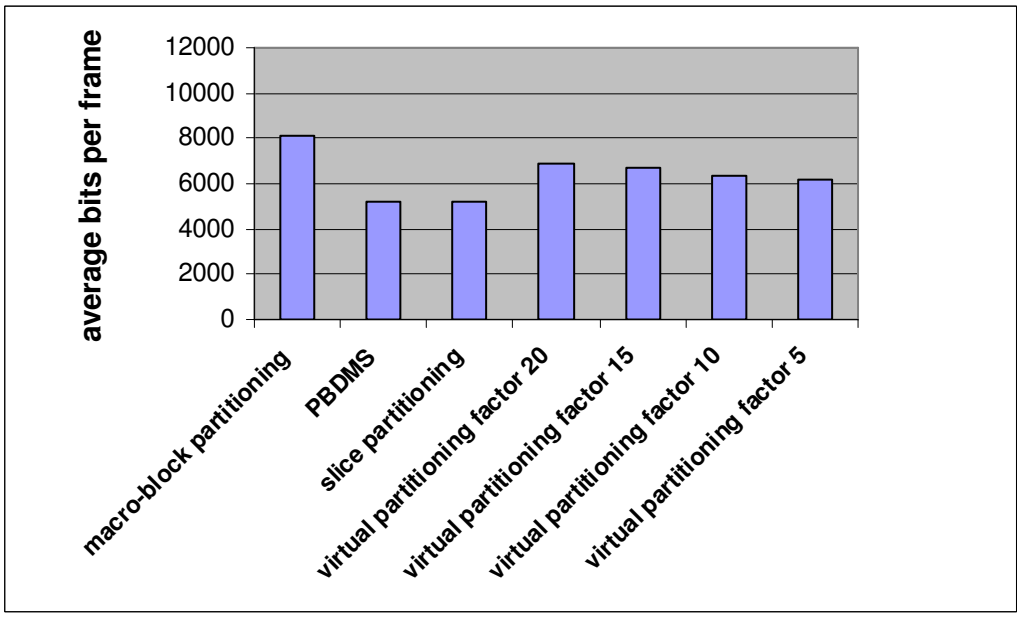


Figure 9-15. Redundancy imposed by existing methods and virtual partitioning on 'carphone' test clip

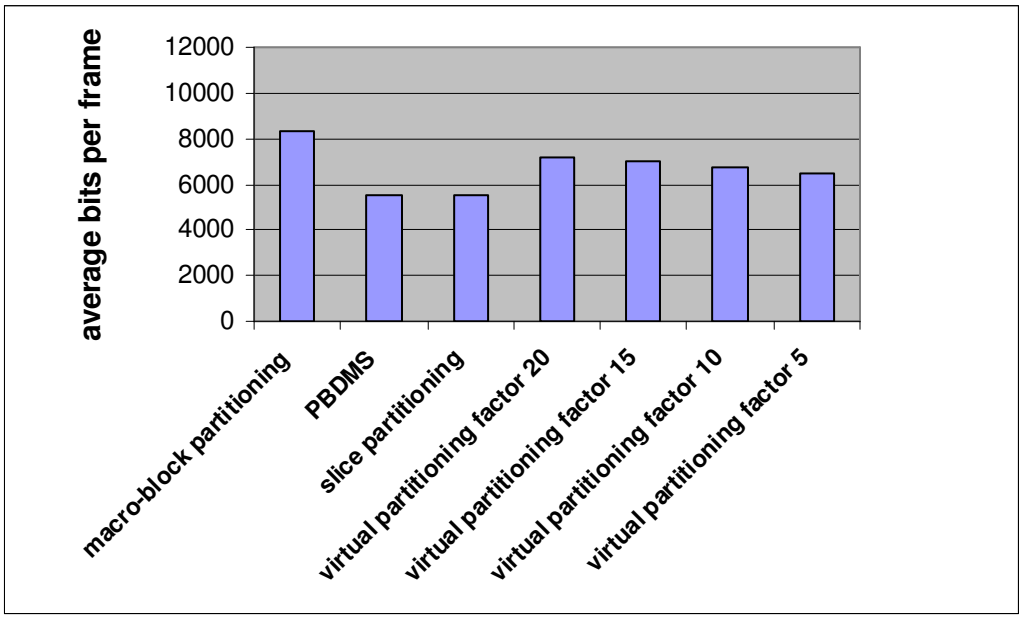


Figure 9-16. Redundancy imposed by existing methods and virtual partitioning on 'Suzie' test clip

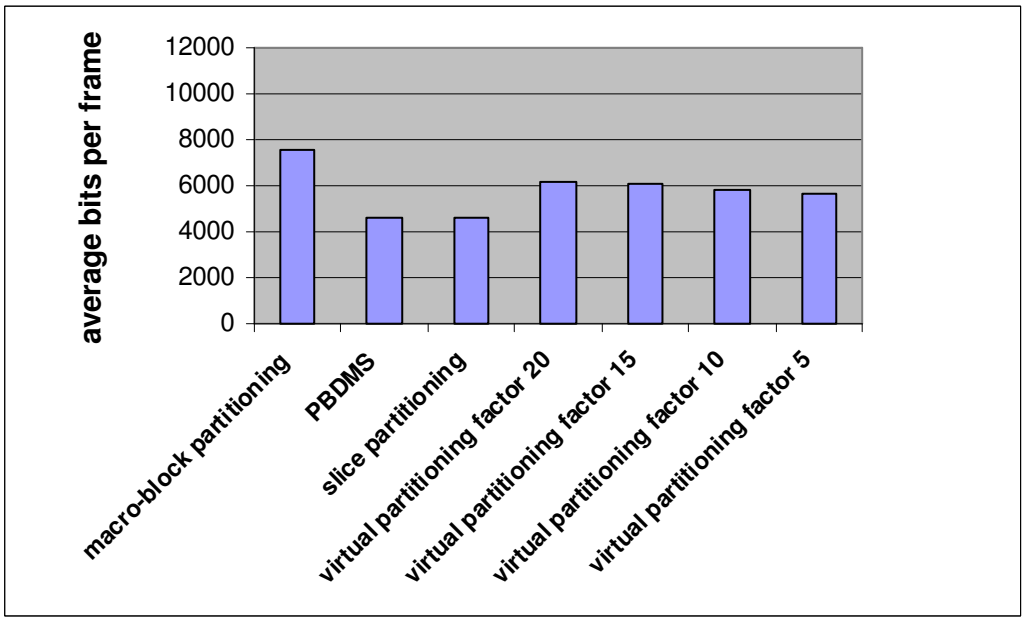


Figure 9-17. Redundancy imposed by existing methods and virtual partitioning on 'salesman' test clip

As the multiplicative factor is reduced, the amount of redundancy decreases gradually, but there is an increase in the number of iterations required for the decoder to regain synchronisation with the bitstream. For example, if a macro-block of 200 bits is erroneous, the decoder will require 10 iterations to reach the macro-block boundary of the next decodable macro-block, if the multiplicative factor is '20', whereas it will require 40 iterations, if the multiplicative factor is '5'. This section explained the method of virtual partitioning, in which the invisible markers are inserted without the physical presence of any bits in the bitstream, but the redundancy is still imposed due to padding. The next section presents the simulation results of the 'modified rate matching' method, in which the virtual partitioning method is achieved without any increase in the bitrate.

9.5. Simulation Results of Virtual Partitioning by Modified Rate Matching Method

The 'modified rate matching' varies the quantisation parameter for achieving virtual partitioning without any redundancy bits. The rate matching algorithm explained in the MPEG-4 standard was used and the multiplicative factor was set to 10. The QP was varied to roll the bits back to the previous location that satisfies the multiplicative rule, by increasing the QP. The errors were imposed in the same location as the virtual partitioning method.

Figures 9-18, 9-19, 9-20 and 9-21 show the performance comparison of the proposed method with the existing methods. Figures show that there is a slight drop in the quality of decoded video with the modified rate matching in comparison with the virtual partitioning. This fractional reduction in PSNR is due to increase in the QP, but the proposed method still outperforms the existing methods. The reduction in PSNR is higher with foreman sequence than other sequences due to extensive motion.

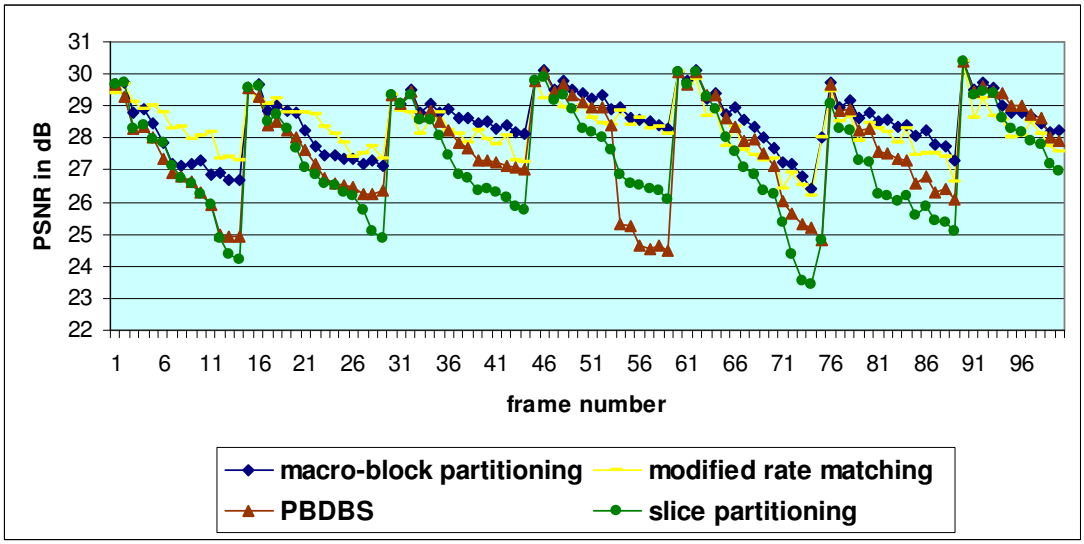


Figure 9-18. Comparison of resync marker methods and modified rate matching on 'foreman' test clip

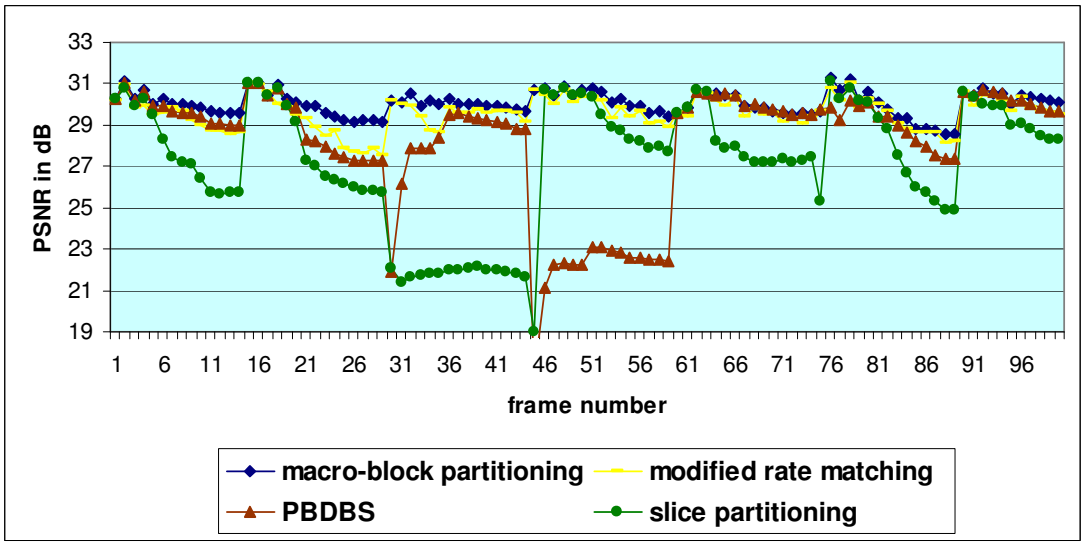


Figure 9-19. Comparison of resync marker methods and modified rate matching on 'carphone' test clip

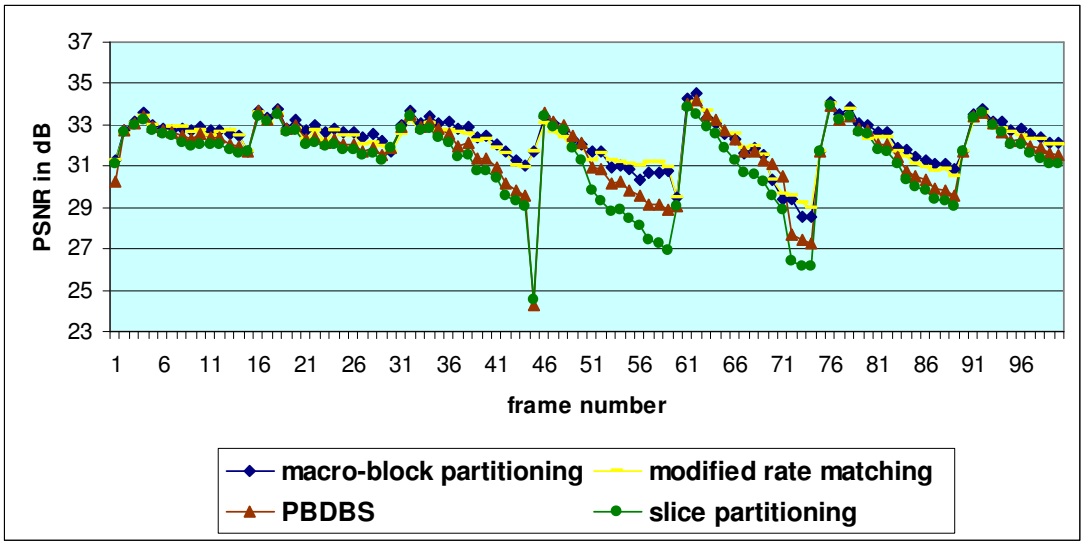


Figure 9-20. Comparison of resync marker methods and modified rate matching on 'Suzie' test clip

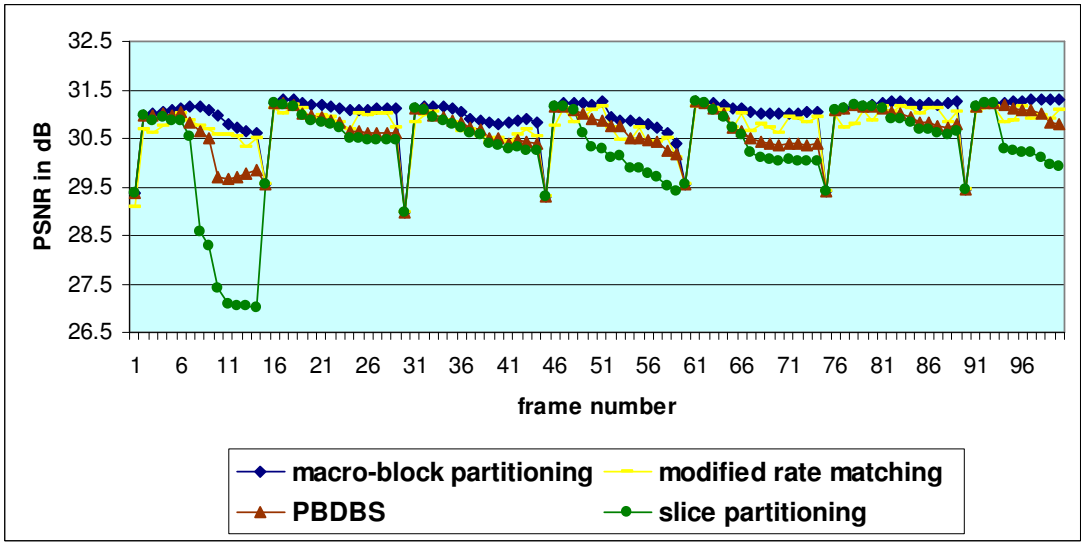


Figure 9-21. Comparison of resync marker methods and modified rate matching on 'salesman' test clip

Figures 9-22, 9-23, 9-24 and 9-25 show the redundancy imposed by the various resync methods. The redundancy imposed by the modified rate matching method is approximately equal to the redundancy imposed by the slice partitioning method, but the performance is as good as the macro-block partitioning method.

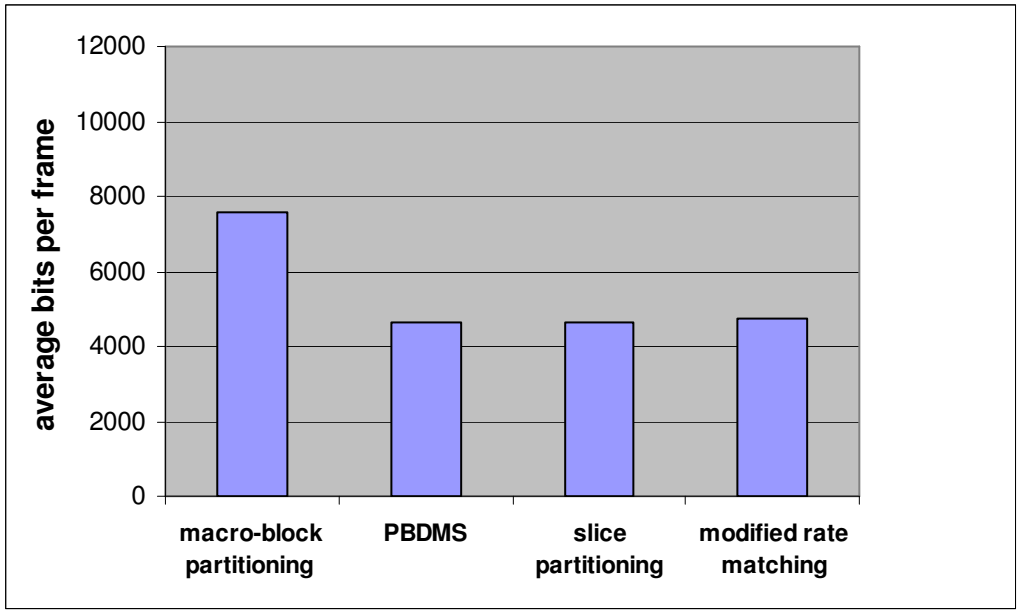


Figure 9-22. Redundancy imposed by existing methods and modified rate matching on 'foreman' test clip

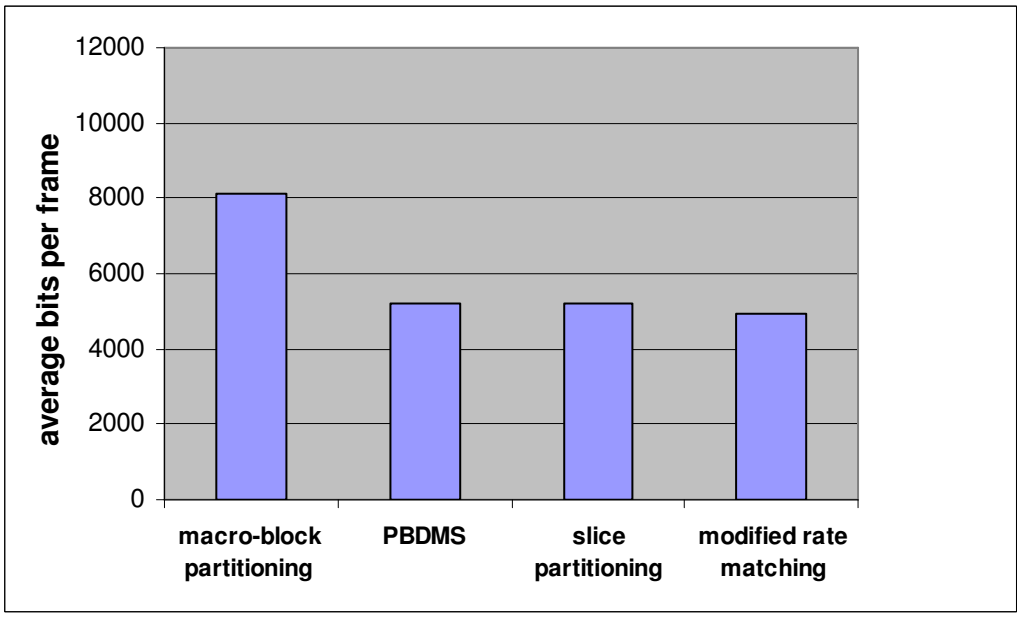


Figure 9-23. Redundancy imposed by existing methods and modified rate matching on 'carphone' test clip

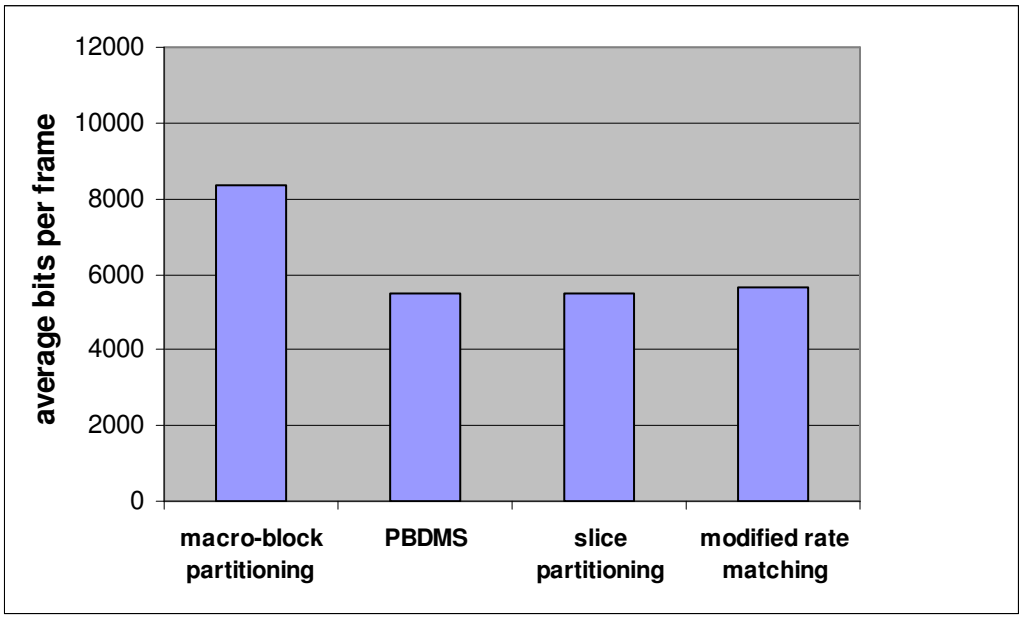


Figure 9-24. Redundancy imposed by existing methods and modified rate matching on 'Suzie' test clip

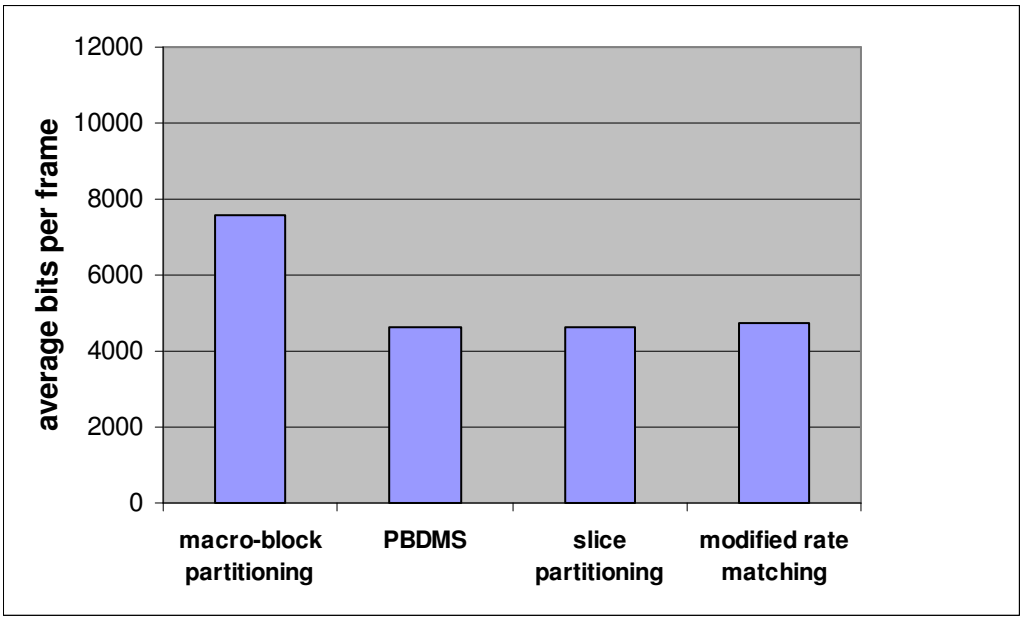


Figure 9-25. Redundancy imposed by existing methods and modified rate matching on 'salesman' test clip

9.6. Summary

The simulation results prove that a better performance can be achieved by adding the redundancy bits to the source coding layer rather than to the channel coding layer. The above observation holds true only when the channel is characterised by bursty errors and the application data exhibits statistical dependency. It was explained in Chapter 7 that the amount of redundancy can be reduced by optimising the source error resilient methods without any degradation in performance. The resync markers are replaced by invisible markers in the virtual partitioning method. The simulation results show that the proposed method achieves the same performance as the macro-block partitioning method, while at the same time reducing the redundancy bit consumption by approximately 30%. Finally, the modified rate matching algorithm adapts the allocation of the bits to the macro-blocks during compression in line with the 'multiplicative rule' and facilitates invisible markers in the bitstream without any redundant bits with a fractional reduction in the video quality.

10. Conclusions

10.1. Discussions

Video has become part of everyday life; from televisions to hand-held devices, the penetration of the technology is very wide. The end-to-end architecture of the video coding system is extremely complex. A video stream must travel through a complicated hierarchy of layers, each of which is driven by a different protocol, before reaching its destination. For a consumer located at the receiving end, the processes running in the background are not visible. Consumers paying a premium to the broadcasters expect high quality video in return. Thus, it has become necessary for the industry to maintain the video quality at a very high standard by resolving any of the problems that will affect the final display quality of the video. Also, the solutions proposed must be compatible and fast enough to be implemented in real time to quickly resolve the quality issues.

The original objective of the thesis was to address the error quality issues that occur in the different layers of the video protocol stack, and subsequently rectify the errors as necessary to improve the picture quality suitable for practical implementation. This thesis is unique for the following reasons: -

- The broad investigation allowed there to be an accurate projection of the influence of an error occurring in a specific layer of the protocol stack to the overall video quality.
- The proposed algorithms were mathematically modelled and designed for commercial implementation. Hence, implemented for practical real world commercial situations.

The issues investigated include:-

- a new problem that occurs in the editing layer of the protocol stack. The solution to the problem is a major contribution to both the broadcasting industry and this thesis.

- a new look at an established transmission layer problem, which has been an active research area for many years.

10.2. Key Contributions

10.2.1. Editing Layer Issues

Videos can be classified into three types, namely, interlaced, progressive and pulldown, based on their origin. The convergence of various video protocols led to hybrid videos, in which videos from different origins coexisted in the same stream. Broadcasters like Google, MTV, Disney and Microsoft sent their erroneous video clips with visual artefacts to Tektronix for analysis. The broadcasters could not fully understand what was causing the problems, as routine quality checks on the bitstream did not identify any errors. This implied that the video file was not corrupt and, probably, something else was wrong. A considerable amount of time was spent in analysing the errors, as the cause was unknown to the broadcasting industry.

Detailed scrutiny of the clips by repeated visual inspection in both progressive and interlaced displays identified the root cause of the problem to be ‘field reversal’ and ‘mixed pulldown’. A significant amount of time was spent liaising with the broadcasters to discover the origin of the problem, which pointed to the editing houses where unskilled workers ignored the field order while editing the video streams captured by different standards. It was also clear that this mistake has been happening in the editing houses for a long time and there are numerous video streams in the market, which makes the process of rectifying the errors by manual eyeballing impossible. Documenting the field reversal and mixed pulldown problems was a major part of the investigation. This resulted in new algorithms being designed to resolve the issues and these were filed as patents. These patents can be considered to be a contribution to the body of knowledge and to the broadcasting industry as these mistakes can now be rectified.

Several algorithms were developed to solve the editing layer issues. Three metrics, ‘convergence ratio’, ‘gradient deviation ratio’ and ‘cluster ratio’ were

proposed to quantify inter-field motion, as they are fundamental components for identifying different frame types. The unique nature of the convergence ratio metric is that it has a constant threshold cut-off and a static dynamic range regardless of the resolution and quality of the image. The unique nature of the gradient deviation ratio is that it emulates the human eye and quantifies the inter-field motion based on human perception. Additionally, the cluster filter offers the flexibility of partitioning a frame into blocks by incorporating an adaptive threshold. The foundation of these algorithms has been through careful mathematical modelling rather than the use of heuristic methods. This has resulted in a highly effective pre-processor being designed for use with de-interlacing, inverse telecine, field reversal and mixed pulldown detection algorithms. This has also improved the independence of the methods with regards to varying video stream parameters, such as spatial frequency, temporal frequency, motion, quality and resolution.

The real time field reversal and mixed pulldown methods were capable of automatically detecting and rectifying the errors in a compressed video stream with a flexibility of implementation on both the source encoder and the decoder ends. This was achieved by incorporating novel methods like variable window size, interpolation, objective correlation metrics, optical flow metric, strip search and moving window. The exhaustive process of manual inspection of hours and hours of video can now be replaced by the proposed system, which can restore the integrity of the video automatically. The ability of the proposed methods to perform in real time has been proven by integrating the algorithms with ‘Cerify’, a video quality control manufactured by Tektronix and made commercially available.

10.2.2. Transmission Layer Issues

Many researchers have investigated improving the robustness of the mobile multimedia, but the methods proposed are still theoretical. This is because of the restrictions on modifications of the standardised protocols. The key reason for the methods being theoretical is their location in the protocol stack. If the methods had been designed to be implemented in the application or source coding layer of the protocol stack, they would have been more real-time friendly. A critical analysis of offering error resilience from the source coding layer and the channel coding layer

was presented. Since the mobile multimedia communication differs from its traditional counterparts in the nature of its application (statistically dependent) and its channel error characteristics (bursty), the investigation has revealed that the errors may be effectively contained in the source coding layer rather than in the channel coding layer. Subsequently, the amount of redundancy required for accomplishing error correction from the source coding layer can be dramatically decreased by using smart optimisation methods, such as 'data hiding', without degrading the performance. It is established that Shannon's source coding theorem of redundancy addition can be achieved from any layer of the protocol stack, and the process is very efficient in the source coding layer with mobile multimedia communications.

One of the smart optimisation methods presented was 'virtual partitioning' of the 'resync marker'. The resync marker that offers resynchronisation to the bitstream in the event of errors by adding redundancy to the bitstream is optimised in such a way that the same functionality is accomplished without the physical presence of any bits in the bitstream. The argument that the increase in the number of resync markers will increase the error resilience is disproved and the flexibility of the source coding layer is proved by 'virtual partitioning' being accomplished with no extra redundancy using a 'modified rate matching' algorithm. The design process has been confined within the application layer of the protocol stack which implies that no modifications in the protocol stack will be required while implementing the algorithms in real-time.

10.2.3. The Hypothesis

The original hypothesis of the thesis is: -

Regardless of the location and the nature of the occurrence of the errors in the compression video transmission system, a more efficient solution for the problems may be obtained from the source coding layer, as it may offer more flexibility than any other layer in the protocol stack

Though errors occurring in the editing layer could be rectified in the display layer of the protocol stack by manual eyeballing, a more efficient solution was found

in the source coding layer using the inter-field quantifier and correlation methods. An investigation into the current solution of visual inspection revealed that the process was very exhaustive and it is not a feasible solution due to the sheer volume of clips, time constraints and practicality. The proposed methods have been proven to emulate human eye with good precision and automate the image analysis with their performance as good as the manual visual inspection. In addition, though the errors occurring in the transmission layer could be rectified in the channel coding layer, the more efficient solution was found in the source coding layer using source error resilient methods. The investigation into current solution of utilizing channel codes revealed their operating limitations in terms of adaptability with channel variations and lack of transparency with higher layers. The proposed methods have proven to perform better by illustrating maximum performance with a very minimum redundancy. These results prove the primary hypothesis of this thesis.

10.3. Limitations and Future Work

The inter-field quantifiers proposed have been tested with field reversal and mixed pulldown algorithms as a pre-processor module. The quantifiers are designed as independent modules and can act as a pre-processor for any higher layer algorithm. The inter-field motion estimation plays a major role in many popular algorithms such as de-interlacing and reverse telecine. In the future, the testing of the inter-field quantifier will be extended to de-interlacing algorithms, which is a primary component of most digital displays and performance variation after integration of inter-field quantifier will be analysed. Since accurate quantification of inter-field motion is possible with static thresholds using proposed methods, the current problem of over-smoothing and under-smoothing of an interlaced frame can be rectified. This will result in a significant increase in the overall picture quality of the digital progressive displays.

The subjective quality of a video stream is a very important quality consideration. If a video coding algorithm is fed with perceptual information, the complexity of the higher layer algorithm could be reduced to a large extent. This is because it may not be necessary to process the frames that do not have perceptual impact on the observer. The perceptual methods used in the design of the metric will

be further extended by designing precise video quality metrics that can perform as good as subjective evaluation. The limitation of the gradient deviation ratio in handling the frames with global motion, due to lack of significant object boundaries could be addressed and possible solutions to solve the issue be investigated.

Current television architecture has been designed with both upstream and downstream channel between the broadcaster and the customer. This allows up-gradation of the terminal software to incorporate new features. The 'field reversal' and 'mixed pulldown' algorithms have been tested in real-time by being implemented at the encoder end, but the decoder-centric implementation has not yet been tested in real-time. In the future, the decoder centric architecture can be tested by collaborating with a broadcaster and sending a software update to the set-up boxes. The current approach to resolve the video coding issues has been by treating the errors rather than rectifying them. This approach can be changed by understanding the root cause of the problem and addressing the cause. The editing layer issues occur in the post production houses not in the broadcasting end. If the broadcaster is specific about their requirements to the post production houses, the issues can be easily rectified. The system designed in this thesis detects the field reversal and mixed pulldown errors in the bitstream, but the process of rectifying the errors varies across the editing factories. It would be advantageous to study various processes used in different post production houses; the best method could be identified and recommendations be made to the broadcasters.

The superiority of the source coding methods over channel coding methods has been established for bursty mobile channels, but the true characteristics of the 'bursty' errors is still unclear. This thesis assumes that the errors occur in clusters and there is a reasonable gap (error-free zone) between successive bursts. This has been disputed by many researchers. Some studies claim that the error-free zone still contains errors with a very minimum probability (Gilbert Elliot Model). Whereas, some claim that since the burst errors are caused as a result of the Doppler frequency, which is due to the speed of the mobile terminal, this is unlikely to happen with video communications, as most people prefer to view video in a static environment rather than a moving environment. Unless the exact nature of the error occurring in the mobile channels is known, it is not possible to design error resilient methods with

good accuracy. Two extremes of wireless channel error characteristics are random and bursty. It has been established in this thesis that the source coding methods are effective when the channel is bursty and the channel coding methods are effective when the channel is random. Further, the proposed research could be extended to design superior source error resilient methods for counteracting even the random errors. If this is accomplished, the research will be complete and could be argued that the source coding methods are better regardless of the channel error characteristics.

In the past, much research has been focussed on adapting the mobile protocol stack which was primarily designed for voice communications to suit data communications. The change in the protocol stack has been very gradual due to global standardisation. Though the methods proposed in this thesis can be confined within the source coding layer, which can be implemented commercially by a simple update process. The techniques can be made more real time friendly by confining all the modifications to the encoder end without disturbing the decoder end. In the future, the algorithms proposed may be further investigated to modify the architecture to be strictly encoder centric for dumb decoder terminals. With the advent of DVB-H, the future challenges of wireless multimedia field are uncertain at the moment, as it is expected that the new technology will bring many new advancements. The research will follow up the change in the protocol stack and the methods proposed in this thesis will be modified to show how source coding layer can be adapted to handle the challenges of the new technology.

References

ABRAMS, S., (2009)-last update, http://www.video-pro.co.uk.

ALEXIOU, A.G. and BOURAS, C.J., (2005), "Rate and loss control for video transmission over UMTS using real-time protocols", *13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems,* pp. 333-336.

ALSAQRE, F.E. and BAOZONG, Y., (2003), "Moving object segmentation from video sequences: an edge approach", *4th EURASIP Conference Focused on Video/Image Processing and Multimedia Communication,* **1**, pp. 193-199.

ANDERSON, G.H., (1999), Video editing and post production: a professional guide. *Focal Press*.

ASHBRIDGE, N., (1937), "Television in Great Britain", *Proceedings of the IRE,* **25**(6), pp. 697-707.

BALKANSKI, A., PURCELL, S. and KIRKPATRICK, J., (1994), A system for compression and decompression of video data using discrete cosine transform and coding techniques. *Patent Number 5,341,318*.

BARMADA, B., GHANDI, M.M., JONES, E.V. and GHANBARI, M., (2005), "Prioritized transmission of data partitioned H.264 video with hierarchical QAM", *IEEE Signal Processing Letters,* **12**(8), pp. 577-580.

BAYLON, D.M. and MCKOEN, K.M., (2006), Method and apparatus for detecting field order in interlaced material. *Patent Number 0,139,491*.

BAYLON, D.M. and MCKOEN, K.M., (2008), Method for detecting interlaced material and field order. *Patent Number 7,450,180*.

BAYLON, D.M., (2007). "On the Detection of Temporal Field Order in Interlaced Video Data", *IEEE International Conference on Image Processing,* **6**, pp. 129-132.

BELDA, A., GUERRI, J.C. and PAJARES, A., (2006), "Adaptive error resilience tools for improving the quality of MPEG-4 video streams over wireless channels", *Proceedings of the 32nd EUROMICRO Conference on Software Engineering and Advanced Applications,* pp. 424-429.

BEUKER, R.A. and SHAH, I.A., (1994), "Analysis of interlaced video signals and its applications", *IEEE Transactions on Image Processing,* **3**(5), pp. 501-512.

BOUAZIZI, I. and GUNES, M., (2004), "Distortion-optimized FEC for unequal error protection in MPEG-4 video delivery", *Proceedings of Ninth International Symposium on Computers and Communications*, pp. 615-620.

BRULS, W.H.A. and CIUHU, C., (2005), "Bridging the interlace and progressive controversy using a progressive enhancement stream on top of the interlace stream and a new de-interlace algorithm", *Digest of Technical Papers from International Conference on Consumer Electronics,* pp. 5-6.

CARLE, G. and BIERSACK, E.W., (1997), "Survey of error recovery techniques for IP-based audio-visual multicast applications", *IEEE Network,* **11**(6), pp. 24-36.

CASAVANT, S.D., HURST JR, R.N., PERLMAN, S.S., ISNARDI, M.A. and ASCHWANDEN, F., (1994), Video/film-mode (3:2 pulldown) detector using patterns of two-field differences. *Patent Number 5,317,398*.

CHEN, M.J., HUANG, C.H. and HSU, C.T., (2004), "Efficient de-interlacing technique by inter-field information", *IEEE Transactions on Consumer Electronics,* **50**(4), pp. 1202-1208.

CHEN, T., SHENG, X.Z. and SHAN, B.S., (2007), "Research on wireless video transmission scheme based on MPEG-4", *Jisuanji Yingyong Yanjiu,* **24**(11), pp. 281-282.

CHILDS, I., (1986), Telecine machines. *Patent Number 4630120*.

CHOU, C.H. and CHEN, C.W., (1996), "A perceptually optimized 3-D subband codec for video communication over wireless channels", *IEEE Transactions on Circuits and Systems for Video Technology,* **6**(2), pp. 143-156.

CHRISTOPHER, T.J. and CORREA, C., (1997), Method and apparatus for identifying video fields produced by film sources. *Patent Number 5,689,301*.

CHRYSSOMALLIS, M., (2002), "Simulation of mobile fading channels", *IEEE Antennas and Propagation Magazine,* **44**(6), pp. 172-183.

CONKLIN, G.J., (2006), Automatic deinterlacing and inverse telecine. *Patent Number 0,024,703*.

COOMBS, G.R., ANDREW, D. and MORRIS, O.J., (1996), Identifying film frames in a video sequence. *Patent Number 5,565,998*.

CORREA, C. and SCHWEER, R., (1994), Method and device for film-mode detection. *Patent Number 5,365,273*.

COVER, T.M. and THOMAS, J.A., (2006), Elements of information theory. *Wiley-Interscience.*

DAO, N. and FERNANDO, W., (2003), "Channel coding for H. 264 video in constant bit rate transmission context over 3G mobile systems", *Proceedings of the2003 International Symposium on Circuits and Systems*, pp. 114-117.

DE HAAN, G. and BELLERS, E.B., (1998), "Deinterlacing - an overview", *Proceedings of the IEEE,* **86**(9), pp. 1839-1857.

DENG, Z., GUO, Y., GU, X., CHEN, Z., CHEN, Q. and WANG, C., (2008), "A comparative review of aspect ratio conversion methods", *Proceedings of International Conference on Multimedia and Ubiquitous Engineering*, pp. 114-117.

DI RUBERTO, C. and DEMPSTER, A., (2000), "Circularity measures based on mathematical morphology", *Electronics Letters,* **36**(20), pp. 1691-1693.

DING, Y., LU, S. and SHI, L., (2006), "A novel de-interlace based on spatio-temporal weight adaptive and edge-directed interpolation" *8th International Conference on Signal Processing*.

DOGAN, S., CELLATOGLU, A., UYGUROGLU, M., SADKA, A.H. and KONDOZ, A.M., (2002), "Error-resilient video transcoding for robust internetwork communications using GPRS", *IEEE Transactions on Circuits and systems for video technology,* **12**(6), pp. 453-464.

DUFAUX, F. and MOSCHENI, F., (1995), "Motion estimation techniques for digital TV: a review and a new contribution", *Proceedings of the IEEE,* **83**(6), pp. 858-876.

ELANGOVAN, P., LUO, G., HARDING, P. and LAWDAY, G., (2008), "Motion vector smoothing algorithm for robust wireless multimedia communications", *4th IEEE International Conference on Circuits and Systems for Communications,* pp. 466-470.

ELANGOVAN, P., LUO, G. and LAWDAY, G., (2007), "Source based error protection for wireless video by motion vector sharing and residual garbaging", *15th International Conference on Digital Signal Processing*, pp 463-466.

ELANGOVAN, P., LUO, G. and LAWDAY, G., (2007), "Structurally efficient video codec for wireless mobile applications", *Fourth International Conference on Information Technology,* pp.190-195.

ELANGOVAN, P., LUO, G. and LAWDAY, G., (2006), "Optimizing video codecs for mobile multimedia applications", *3rd European Conference on Visual Media Production,* pp.182-182.

ELLIOT, E.O., (1963), "Estimates of error rates for codes on burst-noise channels", *Bell Systems Technical Journal,* **42**, pp. 1977-1997.

ELSEN, I., HARTUNG, F., HORN, U., KAMPMANN, M. and PETERS, L., (2001), "Streaming technology in 3G mobile communication systems", *Computer,* **34**(9), pp. 46-52.

FANG, T. and CHAU, L.P., (2005), "Efficient content-based resynchronization approach for wireless video", *IEEE Transactions on Multimedia,* **7**(6), pp. 1021-1027.

FRITCHMAN, B., (1967), "A binary channel characterization using partitioned Markov chains", *IEEE Transactions on Information Theory,* **13**(2), pp. 221-227.

GALLANT, M. and KOSSENTINI, F., (2001), "Rate-distortion optimized layered coding with unequal error protection for robust internet video", *IEEE Transactions on Circuits and Systems for Video Technology,* **11**(3), pp. 357-372.

GAO, S. and TU, G., (2003), "Early resynchronization, error detection and error concealment for reliable video decoding", *Proceedings of International Conference on Communication Technology*, **2**, pp. 1133-1136.

GAO, S.S. and TU, G.F., (2003), "Robust H. 263 video transmission using partial backward decodable bit stream (PBDBS)", *IEEE Transactions on Circuits and Systems for Video Technology,* **13**(2), pp. 182-187.

GARDNER, L.J. and SCOGGINS, D.H., (1991), Closed-loop post production process. *Patent Number 5,051,845*.

GENOVA, T. and LEVY, M.B., (2001)-last update, television history-the first 75 years [Homepage of http://www.tvhistory.tv/], [Online].

GHANBARI, S. and BOBER, M.Z., 2002. "A cluster based method for the recovery of the lost motion vectors in video coding", *International Workshop on Mobile and Wireless Communications Network,* pp. 583-586.

GILBERT, E.N., (1960), "Capacity of a burst-noise channel", *Bell Systems Technical Journal,* **39**, pp. 1253-1265.

GIROD, B and Farber, N., (1999), "Feedback-based error control for mobile video transmission", *Proceedings of the IEEE,* **87**(10)*,* pp. 1707-1723.

GOVE, R.J., MEYER, R.C. and MARSHALL, S.W., (1995), Film-to-video format detection for digital television. *Patent Number 5,398,071*.

HALONEN, T., ROMERO, R.G. and MELERO, J., (2003), GSM, GPRS and EDGE performance: evolution towards 3G/UMTS. *Wiley*.

HANZO, L., CHERRIMAN, P. and STREIT, J., (2007), Video compression and communications: from basics to H. 261, H. 263, H. 264, MPEG 4 for DVB and HSDPA-style adaptive turbo-transceivers. *Wiley-IEEE Press*.

HASKELL, B.G., PURI, A. and NETRAVALI, A.N., (1996), Digital video: an introduction to MPEG-2. *Kluwer Academic Publishers*.

HATA, M. and MASAHARU, (1993), "The Indoor Radio Propagation Loss in Land Mobile Services", *IEEE Transactions on Vehicular Technology,* **81**(7), pp. 943-968.

HOLMA, H. and TOSKALA, A., (2002), Wcdma for Umts. *John Wiley & Sons*.

HUI, Y., GOH, K. and IT, P., (2000), Method and apparatus for interlaced/non-interlaced frame determination, repeat-field identification and scene-change detection. *Patent Number EP1,163,801.*

HUI, Y.W.L., (2005), Progressive/interlace and redundant field detection for encoder. *Patent Number 6,870,568.*

HWANG, T.H., JOO, I.H. and CHOI, K.H., (2004), "An indexing method for spatial object in video using image processing and photogrammetry", *Proceedings of International Geoscience and Remote Sensing Symposium,* **7***,* pp. 4394-4397.

IRETON, M.A. and XYDEAS, C.S., (1991), "Classification of shape for content retrieval of images in a multimedia database", *Sixth International Conference on Digital Processing of Signals in Communications,* pp. 111-116.

ISO/IEC 11172-2, (1993), *Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1,5 Mbit/s - Part 2 : Video.*

ISO/IEC 13818-2, (2000), *Information Technology - Generic Coding of Moving Pictures and Associated Audio Information : Video.*

ISO/IEC 14496-10, (2008), *Information Technology - Coding of Audio Visual Objects : Advanced Video Coding.*

ISO/IEC 14496-2, (2001), *Information Technology - Coding of Audio Visual Objects : Video.*

ISO/IEC 15444-3, (2007), *Information Technology-JPEG 2000 Image Coding System : Motion JPEG 2000.*

ISO/IEC 21000-1, (2004), *Information technology-Multimedia Framework : Vision Technologies and Strategies.*

ITU-T H.261, (1993), *Video Codec for Audio Visual Services at p x 64 kbits.*

ITU-T H.263, (2005), *Video Coding for Low Bit Rate Communication.*

JACK, K., (2004), Video demystified: a handbook for the digital engineer. *Newnes*.

JAN, Y.H. and LIN, D.W., (2002), "Extraction of video objects by combined motion and edge analysis", *IEEE International Symposium on Circuits and Systems,* **5**, pp. 677-680.

JAYANT, N., JOHNSTON, J. and SAFRANEK, R., (1993), "Signal compression based on models of human perception", *Proceedings of the IEEE,* **81**(10), pp. 1385-1422.

JOHANNESSON, R. and ZIGANGIROV, K.S., (1999), Fundamentals of convolutional coding. *Wiley-IEEE Press*.

KARNER, W., (2007), *Link Error Analysis and Modelling for Cross-Layer Design in UMTS Mobile Communications Networks*, PhD Thesis, Vienna University of Technology.

KEATING, S.M. and RICHARDS, J.W., (1995), Method and apparatus for processing an input 60 field/second video signal generated by 3232 pulldown to produce an output video signal. *Patent Number 5,446,497*.

KILDAY, J., PALMIERI, F. and FOX, M.D., (1993), "Classifying mammographic lesions using computerized image analysis", *IEEE Transactions on Medical Imaging,* **12**(4), pp. 664-669.

KIM, Y.T., KIM, S.H. and PARK, S.W., (2002), "Motion decision feedback deinterlacing algorithms", *IEEE International Conference on Image Processing*, pp. 24-28.

KODIKARA, C., WORRALL, S., KONDOZ, A.M. and FABRI, S.N., (2004), "Link adaptation for streaming video in EGPRS mobile networks", *59th IEEE Vehicular Technology Conference*, pp. 2763-2767

KURCEREN, B. and MODESTINO, J.W., (2000), "A joint source-channel coding approach to network transport on digital video", *Proceedings of Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, **2**, pp. 716-726

LEE, D.H., (2008), "A new edge-based intra-field interpolation method for deinterlacing using locally adaptive-thresholded binary image, *Digest of Technical Papers from International Conference on Consumer Electronics,* pp.1-2.

LEE, G.G., HSIN-TE LI, M.J.W. and LIN, H.Y., (2007), "Motion adaptive deinterlacing via edge pattern recognition", *International Symposium on Circuits and Systems,* pp. 2662-2665.

LEE, H., SEONG, D. and LEE, K., (2005), "Design and implementation of efficient error concealment algorithm using adaptive selection of adjacent motion vectors and

data hiding", *Proceedings of 7th International Workshop on Enterprise networking and Computing in Healthcare Industry,* pp. 303-306.

LEE, J. and DICKINSON, B.W., (1994), "Temporally adaptive motion interpolation exploiting temporal masking in visual perception", *IEEE Transactions on Image Processing,* **3**(5), pp. 513-526.

LEE, S.H., KIM, J.K. and LTD, I.C., (2001), "Optimisation-based placement of resynchronisation markers for error robust video transmission", *Electronics Letters,* **37**(6), pp. 348-350.

LI, R., ZENG, B. and LIOU, M.L., (2000), "Reliable motion detection/compensation for interlaced sequences and its applications to deinterlacing", *IEEE Transactions on Circuits and Systems for Video Technology,* **10**(1), pp. 23.

LIU, D. and CHEN, T., (2008), "DISCOV: a framework for discovering objects in video", *IEEE Transactions on Multimedia,* **10**(2), pp. 200-208.

LIU, Y., (2007), Method for upgrading software or content of terminal device based on digital TV data broadcast. *Patent Number 121,679*.

LO, A., HEIJENK, G. and NIEMEGEERS, I., (2005), "Evaluation of MPEG-4 video streaming over UMTS/WCDNL4, dedicated channels", *Proceedings of First International Conference on Wireless Internet,* pp. 182-189.

LONGLEY, A.B. and RICE, P.L., (1968), "Prediction of tropospheric radio transmission loss over irregular terrain; a computer method", *ESSA Technical Report,* **79**(67).

LU, X., TOURAPIS, A.M., YIN, P. and BOYCE, J., (2005), "Fast mode decision and motion estimation for H. 264 with a focus on MPEG-2/H. 264 transcoding", *IEEE International Symposium on Circuits and Systems,* pp. 1246-1249.

MA, Q., ZHOU, H., YU, Q. and KONG, R., (2005), "Realizing MPEG4 video transmission based on mobile station over GPRS", *Proceedings of Wireless Communications, Networking and Mobile Computing,* **2**, pp. 1287-1290.

MACHIDA, Y., MORIO, M. and KIHARA, N., (1979), "Simplified television standards converter as a video tape reproduction system", *IEEE Transactions on Consumer Electronics,* **25(1)**, pp. 45-49.

MALLAT, S., (2006), Super Resolution Bandlet Upconversion for HDTV.

MANJUNATH, B.S., SALEMBIER, P. and SIKORA, T., (2002), Introduction to MPEG-7: multimedia content description interface. *Wiley*.

MARTIN, A. and SMITH, M., (1995), Method and device for film-mode detection and field elimination, *Patent Number 5,452,011*.

MATSUBARA, S. and IT, P., (2008), Pull-down signal detecting apparatus, pull-down signal detecting method, and video-signal converting apparatus. *Patent Number EP1919211A2.*

MENG, J. and CHANG, S.F., (1997), "CVEPS-a compressed video editing and parsing system", *Proceedings of the fourth ACM international conference on Multimedia,* pp. 43-53.

MOCCAGATTA, I., SOUDAGAR, S., LIANG, J. and CHEN, H., 2000. "Error-resilient coding in JPEG-2000 and MPEG-4", *IEEE Journal on Selected Areas in Communications,* **18**(6), pp. 899-914.

MOREIRA, J.C. and FARRELL, P.G., (2006), Essentials of error-control coding. I*Wiley.*

MURCHING, A.M., NAVEEN, T., JASINSCHI, R.S. and TABATABAI, A., (2005), Kalman tracking of color objects. *Patent Number 6,917,692*.

MURPHY, R.L., (1989), Video discrimination between different video formats. *Patent Number 4,860,098.*

MUSTAFA, A. and SETHI, I., (2005), "Detecting retail events using moving edges", *IEEE Conference on Advanced Video and Signal Based Surveillance,* pp. 626-631.

NAVARRO, A., (2002), "Technical aspects of European digital terrestrial television", *Proceedings of MELECON Mediterranean Electrotechnical Conference,* pp. 2-6.

NETRAVALI, A.N. and PRASADA, B., (1977), "Adaptive quantization of picture signals using spatial masking", *Proceedings of the IEEE,* **65**(4), pp. 536-548.

NICOLAS, M., ROUSSEL, J., CRETE, F. and GRENOBLE, F., (2008), "Metrics to evaluate the quality of motion compensation systems in de-interlacing and up-conversion applications", *Digest of Technical Papers from International Conference on Consumer Electronics*, pp.1-2.

O'DONOVAN, P., (2006), "Goodbye, CRT", *IEEE Spectrum,* **43**(11), pp. 38-43.

OH, H.S., KIM, Y., JUNG, Y.Y., MORALES, A.W. and KO, S.J., (2000), "Spatio-temporal edge-based median filtering for deinterlacing", *Digest of Technical Papers from International Conference on Consumer Electronics*, pp. 52-53.

OTANI, K., DAIKOKU, K. and OMORI, H., (1981), "Burst error performance encountered in digital land mobile radio channel", *IEEE Transactions on Vehicular Technology,* **30**(4), pp. 156-160.

OZGEN, M. and LIM, K.W., (2006), Interlaced-to-progressive scan conversion based on film source detection. *Patent Number 7,075,581.*

PANG, D.H., PANG, S.H. and JI, S., (2004), "Spiral intra macroblock refresh with motion vector restriction for low bit-rate video telephony over a 3G network", *IEEE Transactions on Consumer Electronics,* **50**(4), pp. 1038-1043.

PARK, C.S., YE, J. and LEE, S.U., (1994), "Lost motion vector recovery algorithm", *IEEE International Symposium on Circuits and Systems,* **3**, pp. 223-232.

PARK, M.K., KANG, M.G., NAM, K. and OH, S.G., (2003), "New edge dependent deinterlacing algorithm based on horizontal edge pattern", *IEEE Transactions on Consumer Electronics,* **49**(4), pp. 1508-1512.

PATHAK, B.H., CHILDS, G. and ALI, M., (2005), "UEP implementation for MPEG-4 video quality improvement on RLC layer of UMTS", *Electronics Letters,* **41**(13), pp. 733-735.

PÄTZOLD, M., (2002), Mobile fading channels. *Wiley*.

PELLEGRINI, V., BACCI, G. and LUISE, M., (2008), "Soft-DVB, a fully software, GNU radio based ETSI DVB-T modulator", *5th Karlsruhe Workshop on Software Radios.*

PENNEBAKER, W.B. and MITCHELL, J.L., (1993), JPEG still image data compression standard. *Kluwer Academic Publishers*.

QU, Q., PEI, Y., MODESTINO, J.W. and TIAN, X., (2004), "Error-resilient wireless video transmission using motion-based unequal error protection and intraframe packet interleaving", *International Conference on Image Processing* , **2**, pp. 837-840.

QU, Q., PEI, Y. and MODESTINO, J., (2006), "An adaptive motion-based unequal error protection approach for real-time video transport over wireless IP networks", *IEEE Transactions on Multimedia,* **8**(5), pp. 1033-1044.

RANGAN, P.V., SHAH, M. and SHASTRI, V., (2001), Enhanced interactive video with object tracking and hyperlinking. *Patent Number 6,198,833.*

RAPPAPORT, T.S., (2005), Wireless Communications: Principles and Practice. *Prentice Hall*.

REDL, S., OLIPHANT, M.W., WEBER, M.K. and WEBER, M.K., (1995), An introduction to GSM. *Artech House.*

REIBMAN, A., JAFARKHANI, H., WANG, Y. and ORCHARD, M., (2001), Multiple description video using rate-distortion splitting, *International Conference on Image Processing,* **1**, pp. 978-981.

REIMERS, U., (2006), "DVB-The family of international standards for digital video broadcasting", *Proceedings of the IEEE,* **94**(1), pp. 173-182.

REZAEI, M., BOUAZIZI, I. and GABBOUJ, M., (2008), "Joint video coding and statistical multiplexing for broadcasting over DVB-H channels", *IEEE Transactions on Multimedia,* **10**(8), pp. 1455-1464.

RICHARDS, J.W., KRSLJANIN, M. and OZAKI, Y., (1993), Video post-production of material acquired on film*, Patent Number 5,191,427*.

RICHARDSON, I.E.G., (2004), H. 264 and MPEG-4 video compression. *Wiley.*

RICHARDSON, I.E.G., (1999), *Video Coding for Reliable Communications*, PhD Thesis, Robert Gordon University.

RUNGTA, S., TRIPATHI, N., VERMA, A.K. and SHUKLA, A., (2009), "Designing and optimization of codec H.263 for mobile applications", *International Journal of Computer Science and Network Security,* **9**(3), pp. 273-278.

SCHLEGEL, C. and PEREZ, L., (2004), Trellis and turbo coding. *Wiley-IEEE Press.*

SCOTNEY, B.W., COLEMAN, S.A. and HERRON, M.G., (2001), "A systematic design procedure for scalable near-circular Gaussian operators", *International Conference on Image Processing,* **1**, pp. 700-703

SHAH, I.A. and BEUKER, R.A., (1993), Device for splitting a digital interlaced television signal into components. *Patent Number 5,239,377*.

SHANNON, C.E., (1948), "A mathematical theory of communication", *The Bell System technical Journal,* **27**, pp. 623-656.

SRINIVASAN, D., KONDI, L.P. and PADOS, D.A., (2004), "Scalable video transmission over wireless DS-CDMA channels using minimum TSC spreading codes", *IEEE Signal Processing Letters,* **11**(10), pp. 836-840.

STOCKHAMMER, T. and BYSTROM, M., (2004), "H. 264/AVC data partitioning for mobile video communication", *International Conference on Image Processing,* **1**, pp. 545-548.

SU, Y. and SUN, M.T., (2006), "Fast multiple reference frame motion estimation for H.264/AVC", *IEEE Transactions on Circuits and Systems for Video Technology,* **16**(3), pp. 447-452.

SULLIVAN, G.J. and WIEGNAD, T., (2005), "Video compression—From concepts to the H. 264/AVC standard", *Proceedings of the IEEE,* **93**(1), pp. 18-31.

TANGEMANN, M. and RHEINSCHMITT, R., (1994), "Comparison of upgrade techniques for mobile communication systems", *IEEE International Conference on Communications*, pp. 201-205.

TEPE, K.E. and ANDERSON, J.B., (2001), "Turbo coding behavior in Rayleigh fading channels without perfect interleaving", *IEEE Military Communications Conference,* **2**, pp. 1157-1164.

THOMOS, N., ARGYROPOULOS, S., BOULGOURIS, N.V. and STRINTZIS, M.G., (2006), "Robust transmission of H.264/AVC video using adaptive slice grouping and unequal error protection", *IEEE International Conference on Multimedia and Expo,* pp. 593–596.

TREW, T.I.P. and SEELING, G.C., (1994), Method and apparatus for tracking a moving object. *Patent Number 5,280,530.*

TUDOR, P.N., (1995), "MPEG-2 video compression", *Electronics & communication engineering journal,* **7**(6), pp. 257-264.

VAN DER SCHAAR, M., KRISHNAMACHARI, S., CHOI, S. and XU, X., 2003. "Adaptive cross-layer protection strategies for robust scalable video transmission over 802.11 WLANs", *IEEE Journal on Selected Areas in Communications,* **21**(10), pp. 1752-1763.

VANDENDORPE, L. and CUVELIER, L., (1999), "Statistical properties of coded interlaced and progressive image sequences", *IEEE Transactions on Image Processing,* **8**(6), pp. 749-761.

VIDEO TECHNOLOGY MAGAZINE, (2009)-last update, http://www.h263l.com/. Available: http://www.h263l.com/.

WANG, Y. and ZHU, Q.F., (1998), "Error control and concealment for video communication: A review", *Proceedings of the IEEE,* **86**(5), pp. 974-997.

WELLS, A., (1998), MPEG-2 Inverse Telecine Circuit. *Patent Number 5,757,435*.

WINGER, L.L. and JIA, Y., (2006), Progressive video detection with aggregated block SADS. *Patent Number 0188,662.*

WON, K., LEE, K. and LEE, C., (2007), "Effective deinterlacing using selective spatial-temporal interpolation", *2nd International Workshop on Soft Computing Applications.* pp.17-20.

WONG, A. and BISHOP, W., (2006), "Practical content-adaptive subsampling for image and video compression", *Proceedings of the Eighth IEEE International Symposium on Multimedia,* pp. 667-673.

XU, Y. and ZHOU, Y., (2004), "H. 264 video communication based refined error concealment schemes", *IEEE Transactions on Consumer Electronics,* **50**(4), pp. 1135-1141.

YAN, B. and NG, K.W., (2003), "An improved unequal error protection technique for the wireless transmission of MPEG-4 video", *Proceedings of the Joint Conference on Information, Communications and Signal Processing,* **1**, pp. 513-517.

YANG, K.H., KANG, D.W. and FARYAR, A.F., (2001), "Efficient intra refreshment and synchronization algorithms for robust transmission of video over wireless networks", *Proceedings of International Conference on Image Processing,* **1**, pp. 938-941.

YANG, X., LIN, W., LU, Z., ONG, E.P. and YAO, S., (2005), "Motion-compensated residue preprocessing in video coding based on just-noticeable-distortion profile", *IEEE Transactions on Circuits and Systems for Video Technology,* **15**(6), pp. 742-752.

YOO, K., (1998), "Adaptive resynchronisation marker positioning method for error resilient video transmission", *Electronics Letters,* **34**, pp. 2084.

ZHANG, D. and LU, G., (2001), "Segmentation of moving objects in image sequence: A review", *Circuits, Systems, and Signal Processing,* **20**(2), pp. 143-183.

ZHANG, Y., ZHU, C. and YAP, K.H., (2008), "A joint source-channel video coding scheme based on distributed source coding, *IEEE Transactions on Multimedia,* **10**(8), pp. 1648-1656.

Appendix A

This section gives information about the metadata in the compressed video bitstream that provides information on the field order and redundant fields. Five standards that support the interlaced video coding have been considered for this study:-

- MPEG-2
- MPEG-4
- H.264
- VC-1
- DV-25.

MPEG-2 Syntax

```
Sequence Header ( )
{
Group of Picture ( )
{
Picture ( )
{
Slice ( )
{
Macro-block ( )
{
Block layer ( )
 }
 -
 -
 }
```

MPEG-4 Syntax

```
Visual Object Sequence ( )
{
Visual Object ( )
{
Video Object Plane ( )
{
Slice ( )
{
Macro-block ( )
{
Block ( )
}
-
-
}
```

In MPEG-2 and MPEG-4, the *Progressive_Sequence* flag indicates the nature of the coded video frames (progressive, interlaced or mixed frames). The *VOP_Structure* in MPEG-4 and *Picture_Structure* in MPEG-2 convey information

about the type of picture being coded (00-reserved, 01-top_field, 10-bottom_field, and 11-frame_picture). The *Progressive_Frame* flag indicates that the two fields of the frame are interlaced fields, which implies that there is an interval of time between the two. The status of the *Top_Field_First*, *Repeat_First_Field* and *Progressive_Sequence* flags decides the display order and the repetition rate of a particular field in pulldown mode.

H.264 File Format

Negative Abstraction Layer ( )
{
Sequence Parameter Set ( )
{
Picture Parameter Set ( )
{
Coded IDR and Non IDR Picture ( )
{
Slice ( )
{
Macro-block ( )
{
Block ( )
}
-
-
}
-

In H.264, the *Field_Pic_Flag* indicates if the coded slice contains field or frame information. The *Bottom_Field_Flag* indicates the field order of the slice data being coded. The *Pic_Struct* flag indicates the display structure of the frames; the redundant field information is conveyed through this flag as well. The *Bottom_Field_Flag* and the *Field_Pic_Flag* occur in the slice layer, and the *Pic_Struct* flag occurs in the Sequence Parameter Set layer under *VUI* (Video Usability Information). For a hybrid video sequence, the information is updated by

Supplemental Enhancement Information (SEI) with *Payload_Type* with a value of 1. The *NAL_Unit_Type* (Negative Abstraction layer type) for SEI is 6. The *Nal_Unit_Type* and *Pic_Struct* flag are represented in the bitstream using unsigned Exp-Golomb codes.

VC-1 File Format

Picture ( )
{
Slice ( )
{
Macro-block ( )
{
-
-
}

In VC-1, all the flags that convey information on the field structures occur in the Picture layer. Only the advanced profile in VC-1 supports the interlaced video standard. The *Interlace* flag indicates the coding syntax (progressive or interlaced). The *Frame_Coding_Mode* (FCM) indicates the type of the picture following in the video section of the bitstream (0b-progressive, 10b frame-interlace and 11b field-interlace). The *Progressive_Segmented_Frame* (PSF) flag indicates the display process treatment of the decoded frame. The *TFF* (Top field first) and the *RFF* (Repeat frame flag) take the same meaning as that of the MPEG standards. The PULLDOWN flag indicates the presence of the *TFF* and the *RFF* in the bitstream and also signals the repetition rate of the frames in pulldown mode.

DV-25 File Format

Channel ( )
{
DIF Sequence ( )
{
Header, Subcode, Vaux, Audio &Video ( )
{
DIF ( )
{
-
}
-

}

In DV-25, the flags that give information about the field display order are present in the Video Auxiliary Information (VAUX) section. The *Frame/Field* flag (FF) indicates if two consecutive fields are delivered or one field is repeated twice during one frame period. The *First/Second* (FS) flag indicates the display order of the fields (11- field 1 followed by field 2, 10-field 2 followed by field 1, 01- field 1 output twice and 00-field 2 output twice). In DV-25, the top field corresponds to field 2 and the bottom field corresponds to field 1. *Interlaced* Flag (IL) indicates if the video sequence is progressive or interlaced.

Appendix B

The first interpolation method translated into the mathematical Fourier domain is spatial in nature. The spatial interpolation is a process whereby the missing lines are patched by finding the average of the spatial neighbours. The process is kept linear for precise mathematical modelling. The process of shifting the even lines upwards by two lines is given by the following equations:

$$F_E\left(w_x, w_y\right)e^{2jw_y} \tag{A-1}$$

$$\frac{1}{2}F\left(w_x, w_y\right)e^{2jw_y} + \frac{1}{2}F\left(w_x, w_y+\Pi\right)e^{2j(w_y+\Pi)} \tag{A-2}$$

$$\frac{1}{2}F\left(w_x, w_y\right)e^{2jw_y} + \frac{1}{2}F\left(w_x, w_y+\Pi\right)e^{2jw_y} \tag{A-3}$$

Averaging the shifted versions of the frame (4-3-2, A-3) gives the following:

$$\frac{1}{4}F\left(w_x, w_y\right)\left[1+e^{2jw_y}\right] + \frac{1}{4}F\left(w_x, w_y+\Pi\right)\left[1-e^{2jw_y}\right] \tag{A-4}$$

Shifting the average frame down by one line to align with the missing lines of the sub- sampled even line frame (4-3-2) results in Equation (A-6):

$$\left(\frac{1}{4}F\left(w_x, w_y\right)\left[1+e^{2jw_y}\right] + \frac{1}{4}F\left(w_x, w_y+\Pi\right)\left[1-e^{2jw_y}\right]\right)e^{-jw_y} \tag{A-5}$$

$$F_{EA}\left(w_x, w_y\right) = \frac{1}{4}F\left(w_x, w_y\right)\left[e^{-jw_y} + e^{jw_y}\right] + \frac{1}{4}F\left(w_x, w_y+\Pi\right)\left[e^{-jw_y} - e^{jw_y}\right] \tag{A-6}$$

Equation (A-8) represents the spatially interpolated version of the sub-sampled even field.

$$F_{ES}\left(w_x, w_y\right) = F_E\left(w_x, w_y\right) + F_{EA}\left(w_x, w_y\right) \tag{A-7}$$

$$F_{ES}\ (w_x, w_y) = F(w_x, w_y)\left[\frac{1}{2} + \frac{e^{-j w_y}}{4} + \frac{e^{j w_y}}{4}\right] + F(w_x, w_y + \Pi)\left[\frac{1}{2} - \frac{e^{-j w_y}}{4} - \frac{e^{j w_y}}{4}\right] \quad \text{(A-8)}$$

The same process is repeated for patching up the missing lines of the sub-sampled odd line frame (4-3-3). The process of shifting the odd lines upwards by two lines is given by the following equations.

$$F_O\ (w_x, w_y) e^{2 j w_y} \quad \text{(A-9)}$$

$$\frac{1}{2} F(w_x, w_y) e^{2 j w_y} - \frac{1}{2} F(w_x, w_y + \Pi) e^{2 j (w_y + \Pi)} \quad \text{(A-10)}$$

$$\frac{1}{2} F(w_x, w_y) e^{2 j w_y} + \frac{1}{2} F(w_x, w_y + \Pi) e^{2 j w_y} \quad \text{(A-11)}$$

Averaging the shifted versions of the frame (3-3-3, A-11) results in Equation (A-12).

$$\frac{1}{2} F(w_x, w_y)\left[\frac{1}{2} + \frac{e^{2 j w_y}}{2}\right] - \frac{1}{2} F(w_x, w_y + \Pi)\left[\frac{1}{2} - \frac{e^{2 j w_y}}{2}\right] \quad \text{(A-12)}$$

Shifting the average frame down by one line (A-13) to align with the missing lines of the sub-sampled even line frame (4-3-3) results in Equation (A-14).

$$\left(\frac{1}{2} F(w_x, w_y)\left[\frac{1}{2} + \frac{e^{2 j w_y}}{2}\right] - \frac{1}{2} F(w_x, w_y + \Pi)\left[\frac{1}{2} - \frac{e^{2 j w_y}}{2}\right]\right) e^{-j w_y} \quad \text{(A-13)}$$

$$F_{OA}\left(w_x, w_y\right) = \frac{1}{2} F\left(w_x, w_y\right)\left[\frac{e^{-j w_y}}{2} + \frac{e^{j w_y}}{2}\right] + \frac{1}{2} F\left(w_x, w_y + \Pi\right)\left[\frac{e^{-j w_y}}{2} - \frac{e^{j w_y}}{2}\right] \quad \text{(A-14)}$$

Equation (A-14) represents the spatially interpolated version of the sub-sampled odd field.

$$F_{OS}\left(w_x, w_y\right) = F_O\left(w_x, w_y\right) + F_{OA}\left(w_x, w_y\right) \quad \text{(A-15)}$$

$$F_{OS}\left(w_x, w_y\right) = \frac{1}{2} F\left(w_x, w_y\right)\left[1 + \frac{e^{-j w_y}}{2} + \frac{e^{j w_y}}{2}\right] - \frac{1}{2} F\left(w_x, w_y + \Pi\right)\left[1 - \frac{e^{-j w_y}}{2} - \frac{e^{j w_y}}{2}\right] \quad \text{(A-16)}$$

The second interpolation method translated into a mathematical Fourier domain incorporates the temporal coefficients in addition to the spatial coefficients. The spatio-temporal interpolation is a process where the missing lines are patched by finding the average of both temporal and spatial neighbours. The process of shifting the odd lines upwards by one line is given by the following equations.

$$F_O\left(w_x, w_y\right) e^{j w_y} \quad \text{(A-17)}$$

$$\frac{1}{2} F\left(w_x, w_y\right) e^{j w_y} + \frac{1}{2} F\left(w_x, w_y + \Pi\right) e^{j w_y} \quad \text{(A-18)}$$

Three point averaging of the two line shifted spatial coordinates (A-11) and one line shifted temporal coordinates (A-18) is shown in Equation (A-19):

$$F\left(w_x, w_y\right)\left[\frac{1}{6}+\frac{e^{2jw_y}}{6}+\frac{e^{jw_y}}{6}\right]+F\left(w_x, w_y+\Pi\right)\left[\frac{1}{6}+\frac{e^{2jw_y}}{6}+\frac{e^{jw_y}}{6}\right] \quad \text{(A-19)}$$

Equation (A-20) shows the result of shifting the lines down to align with the missing lines of the sub-sampled even line frame.

$$\left(F\left(w_x, w_y\right)\left[\frac{1}{6}+\frac{e^{2jw_y}}{6}+\frac{e^{jw_y}}{6}\right]+F\left(w_x, w_y+\Pi\right)\left[\frac{1}{6}+\frac{e^{2jw_y}}{6}+\frac{e^{jw_y}}{6}\right]\right)e^{-jw_y} \quad \text{(A-20)}$$

$$F_{EST}\left(w_x, w_y\right)=F\left(w_x, w_y\right)\left[\frac{1}{6}+\frac{e^{-jw_y}}{6}+\frac{e^{jw_y}}{6}\right] - F\left(w_x, w_y+\Pi\right)\left[\frac{1}{6}+\frac{e^{-jw_y}}{6}+\frac{e^{jw_y}}{6}\right] \quad \text{(A-21)}$$

The addition of the shifted frame with the interpolated values to the original sub-sampled even line frame is shown in Equation (A-22). The spatio-temporal interpolated even line frame is given by Equation (A-23).

$$F_{EVT}\left(w_x, w_y\right) = F_E\left(w_x, w_y\right) + F_{EST}\left(w_x, w_y\right) \quad \text{(A-22)}$$

$$F_{EVT}\left(w_x, w_y\right)=F\left(w_x, w_y\right)\left[\frac{2}{3}+\frac{e^{-jw_y}}{6}+\frac{e^{jw_y}}{6}\right] + F\left(w_x, w_y+\Pi\right)\left[\frac{1}{3}-\frac{e^{-jw_y}}{6}-\frac{e^{jw_y}}{6}\right] \quad \text{(A-23)}$$

The process is repeated for spatio-temporal interpolation on the odd line frame. Shifting the even line frame by one line upwards results in Equation (A-26):

$$F_E\left(w_x, w_y\right)e^{jw_y} \quad \text{(A-24)}$$

$$\frac{1}{2}F\left(w_x, w_y\right)e^{j w_y} + \frac{1}{2}F\left(w_x, w_y+\Pi\right)e^{j(w_y+\Pi)} \quad \text{(A-25)}$$

$$\frac{1}{2}F\left(w_x, w_y\right)e^{j w_y} - \frac{1}{2}F\left(w_x, w_y+\Pi\right)e^{j w_y} \quad \text{(A-26)}$$

Three point averaging of the one line shifted spatial coordinates (A-11) and two line shifted temporal coordinates (4-5-11) results in Equation (A-27).

$$F\left(w_x, w_y\right)\left[\frac{1}{6}+\frac{e^{2j w_y}}{6}+\frac{e^{j w_y}}{6}\right]-F\left(w_x, w_y+\Pi\right)\left[\frac{1}{6}+\frac{e^{2j w_y}}{6}+\frac{e^{j w_y}}{6}\right] \quad \text{(A-27)}$$

Shifting the lines down to align with the missing lines of the sub-sampled odd line frame results in Equation (A-29).

$$\left(F\left(w_x, w_y\right)\left[\frac{1}{6}+\frac{e^{2j w_y}}{6}+\frac{e^{j w_y}}{6}\right]-F\left(w_x, w_y+\Pi\right)\left[\frac{1}{6}+\frac{e^{2j w_y}}{6}+\frac{e^{j w_y}}{6}\right]\right)e^{-j w_y} \quad \text{(A-28)}$$

$$F_{OST}\left(w_x, w_y\right) = F\left(w_x, w_y\right)\left[\frac{1}{6}+\frac{e^{j w_y}}{6}+\frac{e^{-j w_y}}{6}\right] - F\left(w_x, w_y+\Pi\right)\left[\frac{1}{6}+\frac{e^{j w_y}}{6}+\frac{e^{-j w_y}}{6}\right] \quad \text{(A-29)}$$

The addition of the shifted frame with the interpolated values to the original sub-sampled odd line frame represented by Equation (A-17). The spatio-temporal interpolated odd line frame is given by Equation (A-31).

$$F_{OVT}\left(w_x, w_y\right) = F_O\left(w_x, w_y\right) + F_{OST}\left(w_x, w_y\right) \quad \text{(A-30)}$$

$$F_{OVT}\ \left(w_x\ ,\ w_y\ \right) = F\left(w_x\ ,\ w_y\ \right)\left[\ \frac{2}{3} + \frac{e^{j\,w_y}}{6} + \frac{e^{-j\,w_y}}{6}\ \right]$$
$$+\ F\left(w_x\ ,\ w_y + \Pi\ \right)\left[-\frac{1}{3} + \frac{e^{j\,w_y}}{6} + \frac{e^{-j\,w_y}}{6}\ \right] \qquad \text{(A-31)}$$

Appendix C

P109984EP/MH

A METHOD OF QUANTIFYING INTER-FIELD MOTION IN A VIDEO FRAME

ABSTRACT

A method of quantifying inter-field motion in a video frame (10), the method comprising generating from the video frame (10) a top field (20) and a bottom field (30), interpolating each of the top field (20) and the bottom field (30) so generated to produce interpolated top and bottom field images (40, 50) and comparing the interpolated top and bottom field images (40, 50) to each other to determine a value representative of the amount of inter-field motion present between the top field (20) and the bottom field (30).

The present invention relates to a method of quantifying inter-field motion in a video frame. Video frames can be classified as either progressive or interlaced, depending upon the method used to display them. In a progressive frame, the horizontal lines of pixels that make up the frame are displayed line by line from top to bottom. In contrast, an interlaced frame is created by displaying two fields in turn, one field (known as the top field) containing the top line of the frame and every second subsequent line, and the other field (the bottom field) containing the second line from the top and every second subsequent line, thus including the bottom line of the frame. Interlaced frames rely on the fact that it takes time for the first field of displayed pixels to decay from the display apparatus, during which time the second field is displayed, to create the illusion of a single frame containing all of the lines of pixels.

The fields of an interlaced video frame are captured sequentially, which means that there is a small time delay between the first field to be captured and the second field to be captured. It is possible for the information contained in the source video frame to change in this time interval, and for this reason it is important that the fields of the video frame are displayed in the correct order, as will be explained below.

Interlaced video frames can be described as either “top field first” or “bottom field first”, depending upon which of the fields making up the frame is intended to be displayed first. As there is a small time delay between displaying the first field and displaying the second field, the field intended to be displayed second may contain different information from that contained in the field intended to be displayed first, for example if movement has occurred in the frame in the delay between displaying the first and second fields. Such differences between the field intended to be displayed first and the field intended to be displayed second are known as “inter-field motion”. If fields containing inter-field motion are displayed in an incorrect order, distortion may appear in the displayed frame. In an interlaced display, for example, the video typically becomes juddery or shaky as information appears earlier than it was intended to appear. In a progressive display, the reversal of the fields will not cause such juddery or shaky video, as the fields are put together and displayed at a rate of N frames per second, rather than 2N fields per second, but regardless of the field order, the inter-field motion will lead to combing artefacts, i.e. areas of the frames where rows of lines appear, giving a “combed” appearance.

Many video applications, such as inverse telecine, interlaced to progressive conversion and field dominance detection, require a classification of a video frame as either interlaced or progressive. The amount of inter-field motion in a video frame may be used to indicate whether a video frame is progressive or interlaced. Different methods exist for performing this interlaced/progressive classification, but none of them offers a unique metric which is consistent for different image resolutions, types and quality.

According to a first aspect of the present invention, there is provided a method of quantifying inter-field motion in a video frame, the method comprising generating from the video frame a top field and a bottom field, interpolating the top field and bottom field so generated to produce interpolated top and bottom field images and comparing the interpolated top and bottom field images to each other to determine a value representative of the amount of inter-field motion present between the top field and the bottom field.

The interpolated top field image may be produced by averaging adjacent lines of the top field with a line of the bottom field which is intermediate the adjacent lines of the top field, and the interpolated bottom field image may be produced by averaging adjacent lines of the bottom field image with a line of the top field image which is intermediate the adjacent lines of the bottom field image.

A difference domain frame may be generated by subtracting one of the interpolated top field image and the interpolated bottom field image from the other of the interpolated top field image and the interpolated bottom field image. Values of pixels of the difference domain frame may be scaled by a scaling factor. The scaling factor may be determined according to a display size. Additionally or alternatively, the scaling factor may be determined according to a distance of an observer of a display from the display. A metric may be calculated for a block of pixels of the difference domain frame, the metric being indicative of the amount of inter-field motion present in the block. The metric may be calculated by determining the number of pixels of the block having immediate horizontal and vertical neighbours with a non-zero value. The block may be classified as being progressive or interlaced by comparing the metric to a threshold.

The size of the block may be calculated based on the resolution of the video frame. The threshold may be calculated based on the size of the block. Alternatively, a metric may be calculated for the difference domain frame, the metric being indicative of the amount of inter-field motion present in the difference domain frame.

The metric may be based on a gradient value of the values of the pixels in the difference domain frame. The gradient value may be normalised with the Mean Absolute Deviation of the value of the pixels in the difference domain frame. Embodiments of the invention will now be described, strictly by way of example only, with reference to the accompanying drawings, of which

Figure 1 is a schematic illustration showing top and bottom fields being generated from a video frame; Figure 2 is a schematic illustration showing interpolated top and bottom field images; Figure 3 is a schematic illustration showing subtraction of blocks of pixels of interpolated top and bottom field images; and Figure 4 is a schematic illustration showing an exemplary 3 x 3 pixel block of a difference domain frame.

Referring first to Figure 1, a video frame is shown generally at 10 and comprises horizontal lines 12, 14 of pixels which make up an image. Typically, a frame comprises 625 lines for a European video system (known as the Phase Alternate Line, or PAL, standard) or 525 lines for a US system (known as the "National Television System Committee, or NTSC, standard).

In order to quantify the amount of inter-field interference occurring in the video frame 10, the video frame 10 must be divided into top and bottom fields. The top field, shown generally at 20 in Figure 1, is generated by extracting the top line 12 of pixels from the frame 10, and every second subsequent line of pixels, and storing these lines in the position from which they were extracted from the frame 10 in the top field 20. Similarly, the bottom field, shown generally at 30 in Figure 1, is generated by extracting the second line 14 of pixels, and every subsequent second line of pixels, and storing them in the position from which they were extracted from the frame 10 in the bottom field 30.

The top and bottom fields 20, 30 each contain only half of the information contained in the video frame 10 from which they were generated. Thus, the top and

bottom fields must be interpolated to produce top and bottom field images each containing as much information as the video frame 10, i.e. to amplify the data contained in the top and bottom fields.

Any interpolation method may be used, although it is advantageous to use a method that the inventors term “three-way interpolation”, as this method takes account of the time delay which occurs between the first field to be displayed and the second field to be displayed and of properties of the human visual system (HVS). The “three way interpolation” process is a method based on the principle that the influence of a pixel on the human vision at any point in time is dependent upon what an observer saw in the previous frame (backward temporal masking), what he is about to see in the next frame (forward temporal masking) and the contrast value of the pixel with respect to its neighbours (spatial masking). The “three way interpolation” process also smooths the effect of edges and contours which occur naturally in the frame 10, which may be similar in appearance and characteristics to combing artefacts.

In the “three-way interpolation” method, adjacent lines of pixels in the field to be interpolated are averaged with a line of the other field which is intermediate the lines of the field to be interpolated. Thus, for example, to generate the second line of an interpolated top field image, as shown at 40 in Figure 2, the value of each pixel of the top line 22 of the top field 20 is summed with the value of the corresponding pixel of the second line 24 of the top field 20 and the value of the corresponding pixel of the first line 32 of the bottom field 30. The resulting sum of pixel values is divided by three to obtain an average pixel value, and the “missing” second line of the top field 20 is built up from the average pixel values calculated in this way.

Similarly, to generate the second line of an interpolated bottom field image, as shown at 50 in Figure 2, the value of each pixel of the first line 32 of the bottom field 30 is summed with the value of the corresponding pixel of the second line 34 of the bottom field 30 and the value of the corresponding pixel of the second line 24 of the

top field 20. The resulting sum of pixel values is divided by three to obtain an average pixel value, and the "missing" second line of the bottom field 30 is built up from the average pixel values calculated in this way.

This process is repeated to generate, from the top and bottom fields 20, 30, interpolated top and bottom field images 40, 50, each of which contains as much information as the frame 10 from which the top and bottom fields 20, 30 were generated. A major advantage of the "three way interpolation" process is that it can be used in place of a more time-consuming process of applying image processing algorithms for estimating spatial and temporal masking to the top and bottom fields 20, 30.

The interpolated top and bottom field images 40, 50 are effectively progressive frames which represent the information which can be seen at the time at which each of the top and bottom fields 20, 30 are displayed in an interlaced system.

Once the interpolated top and bottom field images 40, 50 have been generated, they must be compared to each other to determine whether there is any inter-field motion between them, and if so, to quantify the inter-field motion.

If the video frame 10 from which the interpolated top and bottom field images 40, 50 were derived is a true progressive frame, there would be only a very small difference between the interpolated top and bottom field images 40, 50, resulting from noise, compression, interpolation approximation and vertical differences. The top and bottom fields derived from the video frame 10 are not exactly the same even if there is no motion between them, as they each represent different portions of the frame 10.

Comparison of the interpolated top and bottom field images 40, 50 is performed by subtracting luminance values of the pixels of one of the images 40, 50 from luminance values of corresponding pixels of the other of the images 40, 50, to

generate a “difference domain” frame. In a block-based quantifier, this subtraction operation is carried out in relation to corresponding blocks of pixels of each of the images 40, 50 in turn, whereas in a frame-based quantifier, the subtraction is performed on all of the pixels of each image 40, 50 at once.

Figure 3 illustrates a “block-wise” subtraction operation. The block 60 is a block of dimensions 4 pixels by 4 pixels taken from the interpolated top field image 40. Each of the pixels in the block 60 has a luminance value. For example, the pixel 62 has a luminance value of 235.

The block 70 is a block of dimensions 4 pixels by 4 pixels taken from the interpolated bottom field image 50, which has been taken from a position in the interpolated bottom field image 50 which corresponds to the position in the interpolated top field image from which the block 60 was taken. Again, each of the pixels in the block 70 has a luminance value.

The block 80 is a “difference block” which represents the absolute value of the result of the subtraction of block 70 from block 60. Thus in block 80, the value of each pixel represents the difference between the luminance value of the corresponding pixel of frame 60 and the luminance value of the corresponding pixel of frame 70.

The subtraction can be made fine or coarse using a scaling factor. The scaling factor may be determined based on the target application. The influence of inter-field motion on the human eyes is a function of the display size and the observer’s distance from the display. The influence of inter field motion perceived by the observer at a distance of 50 metres is different from what he perceives from a distance of 100 metres. Similarly, the influence of the inter field motion perceived by the observer when viewed in a computer screen is different from what he will perceive when viewing the same image in a high definition flat screen display. Thus, the scaling factor may be determined according to the display size and/or the distance

of the observer from the display. Simple methods like Weber's law or complicated calculations like JND (Just Noticeable Distortion) could be used for efficient estimation of the scaling factor. This process does not influence the quantifying process, it merely affects the quantifier scale and the level of inter-field motion it corresponds to. The block 82 is the difference block 80 scaled by a factor of five, with non-integer results being rounded to the nearest integer. A plurality of difference blocks 80, 82 may be combined to generate the difference domain frame.

Having obtained the difference domain frame, the amount of inter-field motion in the frame 10 from which the difference domain frame is derived can be quantified, and the frame 10 can thus be classified as either progressive or interlaced. In one embodiment of the invention, quantifying the inter-field motion in the frame 10 is performed using a cluster filter, as will be described below. Inter-field motion in a video frame, like many other visual artefacts, follows the clustering principle, that is to say that if inter-field motion artefacts are closely distributed within a frame (i.e. they are clustered), their impact, in terms of how noticeable they are to a person viewing the frame, is greater than if they are widely distributed. The difference domain frame is divided into a plurality of blocks 80, 82 for processing. For each block, the number of pixels having a neighbour of non-zero difference value is estimated or counted. In this way, the distribution of the pixel values in the difference domain frame (widely spaced or clustered) can be established. As non-zero pixel difference values only occur in the difference domain frame where there is a difference between the interpolated top field image 40 and the interpolated bottom field image 50, i.e. where there is inter-field motion, a large number of pixels having neighbours with non-zero difference values is indicative of a large amount of inter-field motion.

For each block 80, 82 of the difference domain frame, a cluster metric is calculated which is indicative of the nature of the distribution of inter-field motion in the frame 10. The cluster metric is calculated by incrementing a counter for each pixel in the block 80, 82 whose immediate horizontal and vertical neighbours are

non-zero. Each pixel has a horizontal coordinate i and a vertical coordinate j. Thus, for each pixel in the block 80, 82, if the value of the pixels at (i - 1, j), (i + 1, j), (i, j - 1) and (i, j + 1) are all greater than zero, the counter is incremented. The cluster metric of the block 80, 82 is the final value of the counter when all of the pixels in the block 80, 82 have been examined.

The cluster metric of a block 80, 82 can be used to classify the block 80, 82 as progressive or interlaced, by comparing it to a threshold. If the cluster metric of the block 80, 82 is lower than the threshold, it can be considered to be progressive, whereas if the cluster metric of the block 80, 82 is higher than the threshold is can be considered to be interlaced. A small transitional range may be present around the threshold value where the block 80, 82 could be considered to be either progressive or interlaced.

The size of the blocks 80, 82 of the difference domain frame for which cluster metrics are calculated must be carefully selected to ensure that the intensity of inter-field motion is not underestimated due to a large number of stationary pixels (i.e. pixels with a value of zero) in the block 80, 82. The block size will be dependent upon the resolution of the frame, and it has been found that the equation

$$\text{Block Size} = 0.66 \times (\text{Frame Resolution})^{0.67}$$

provides good results, although it will be appreciated that other methods could be used to calculate the block size.

The threshold for assessing whether a block 80 is progressive or interlaced can be determined according to the block size, for example using the equation

$$\text{Threshold} = 1.05 \times (\text{Block Size})^{0.6},$$

although again it will be appreciated that other methods for calculating the threshold may be used.

The frame 10 may be classified as progressive or interlaced on the basis of the number of blocks 80, 82 which are classified, according to the cluster filter, as being interlaced or progressive. For example, if over thirty per cent of the blocks 80, 82 are classified as being interlaced, the frame 80 may be classified as interlaced. The results can be effectively interpreted and visualized. For example, a frame containing one block having a very high value and a frame having multiple blocks with a medium value will have the same effect on the viewer.

In an alternative embodiment, a gradient value of the pixels of the difference domain frame is used to quantify the inter-field motion in the frame 10, as will be described below with reference to Figure 4.

The gradient value of a pixel is the directional derivative in x and y directions, and is indicative of the deviation of the value of the pixel from the value of neighbouring pixels. As inter-field motion is characterised by differences between neighbouring pixels, a high gradient value may be indicative of inter-field motion.

Figure 4 shows an exemplary 3 x 3 pixel block 90 of the difference domain frame. Each of the pixels has a horizontal (x) coordinate and a vertical (y) coordinate. Each pixel has an x gradient value, indicative of its deviation from the value of horizontally adjacent pixels, and a y gradient value, indicative of it deviation from the value of vertically adjacent pixels. The x gradient can be calculated using the formula

$G_x = \frac{(P_{(i+1)} - P_i) + (P_i - P_{(i-1)})}{2}$, where P_i is the value of a pixel at horizontal position i.

Similarly, the y gradient can be calculated using the formula

$G_y = \frac{(P_{(j+1)} - P_j) + (P_j - P_{(j-1)})}{2}$, where P_j is the value of a pixel at vertical position j.

Thus, for the central pixel 92 of the block 90 shown in Figure 4,

$G_x = \frac{(6-5\ ) + (5-4)}{2} = 1$, whilst

$G_y = \frac{(2-5) + (5-8)}{2} = -3$

The overall gradient value of the pixel can be determined using the equation

$$G = \frac{(|G_x| + |G_y|)}{2}$$

Thus, for the central pixel 92 of the block 90 of Figure 4

$$G = \frac{(|1| + |-3|)}{2} = 2$$

The gradient value for the entire difference domain frame can be calculated by summing the overall gradient values of the individual pixels and dividing the result by the total number of pixels in the frame, i.e.

$$Grad = \frac{\sum_{i=0}^{m}\sum_{j=0}^{n} G_{i,j}}{mn}$$

Similarly the Mean Absolute Deviation (MAD) of pixel luminance values from the mean of the pixel luminance values can be used to give an indication of the amount of inter-field motion in a frame.

To determine the MAD of the difference domain frame, the following equation may be used

$MAD = \frac{\sum_{i=0}^{m}\sum_{j=0}^{n}\left|P_{i,j} - M\right|}{mn}$, where $P_{i,j}$ is the value of a pixel at position i, j, M is the mean value of the pixels in the frame, m is the number of pixels per horizontal row of the frame and n is the number of pixels per vertical column of the frame.

The gradient value of the entire difference domain frame can be normalised by dividing by the MAD of the difference domain frame, to calculate a Gradient Deviation Ratio, as follows:

$$GDR = \frac{Grad}{MAD}.$$

This normalisation process produces a value for inter-field motion which falls within a uniform range regardless of the characteristics or resolution of the video frame 10. The gradient deviation ratio typically produces a value between 0 and 1 which can be used to quantify inter-field motion in a video frame, and thus to characterise the video frame as progressive or interlaced. A threshold, or a series of thresholds, may be provided for the purpose of progressive/interlaced classification. For example, if the GDR is less than 0.5, the frame may be classified as interlaced. If the GDR is greater than 0.7, the frame may be classified as progressive. If the GDR is between 0.5 and 0.7, the frame cannot accurately be classified as either progressive or interlaced.

The gradient of the difference domain frame may also be calculated using edge detection masks such as Sobel, Prewitt, Canny or Frei-Chen, and the mask used may have an effect on the range of the value or metric produced by the gradient deviation ratio.

Although this method provides satisfactory results, it does not take into account the clustering effect of the human visual system, and thus the results are not as accurate as those produced by the cluster filter embodiment discussed above.

CLAIMS

1. A method of quantifying inter-field motion in a video frame, the method comprising generating from the video frame a top field and a bottom field, interpolating each of the top field and the bottom field so generated to produce interpolated top and bottom field images and comparing the interpolated top and bottom field images to each other to determine a value representative of the amount of inter-field motion present between the top field and the bottom field.

2. A method according to claim 1 wherein the interpolated top field image is produced by averaging adjacent lines of the top field with a line of the bottom field which is intermediate the adjacent lines of the top field, and the interpolated bottom field image is produced by averaging adjacent lines of the bottom field image with a line of the top field image which is intermediate the adjacent lines of the bottom field image.

3. A method according to claim 1 or claim 2 wherein a difference domain frame is generated by subtracting one of the interpolated top field image and the interpolated bottom field image from the other of the interpolated top field image and the interpolated bottom field image.

4. A method according to claim 3 wherein values of pixels of the difference domain frame are scaled by a scaling factor.

5. A method according to claim 4 wherein the scaling factor is determined according to a display size.

6. A method according to claim 4 or claim 5 wherein the scaling factor is determined according to a distance of an observer of a display from the display.

7. A method according to any one of claims 3 to 6 wherein a metric is calculated for a block of pixels of the difference domain frame, the metric being indicative of the amount of inter-field motion present in the block.

8. A method according to claim 7 wherein the metric is calculated by determining the number of pixels of the block having immediate horizontal and vertical neighbours with a non-zero value.

9. A method according to claim 8 wherein the block is classified as being progressive or interlaced by comparing the metric to a threshold.

10. A method according to any one of claims 7 to 9 wherein the size of the block is calculated based on the resolution of the video frame.

11. A method according to claim 10 wherein the threshold is calculated based on the size of the block.

12. A method according to claim 3 wherein a metric is calculated for the difference domain frame, the metric being indicative of the amount of inter-field motion present in the difference domain frame.

13. A method according to claim 12 wherein the metric is based on a gradient value of the values of the pixels in the difference domain frame.

14. A method according to claim 13 wherein the gradient value is normalised with the Mean Absolute Deviation of the value of the pixels in the difference domain frame.

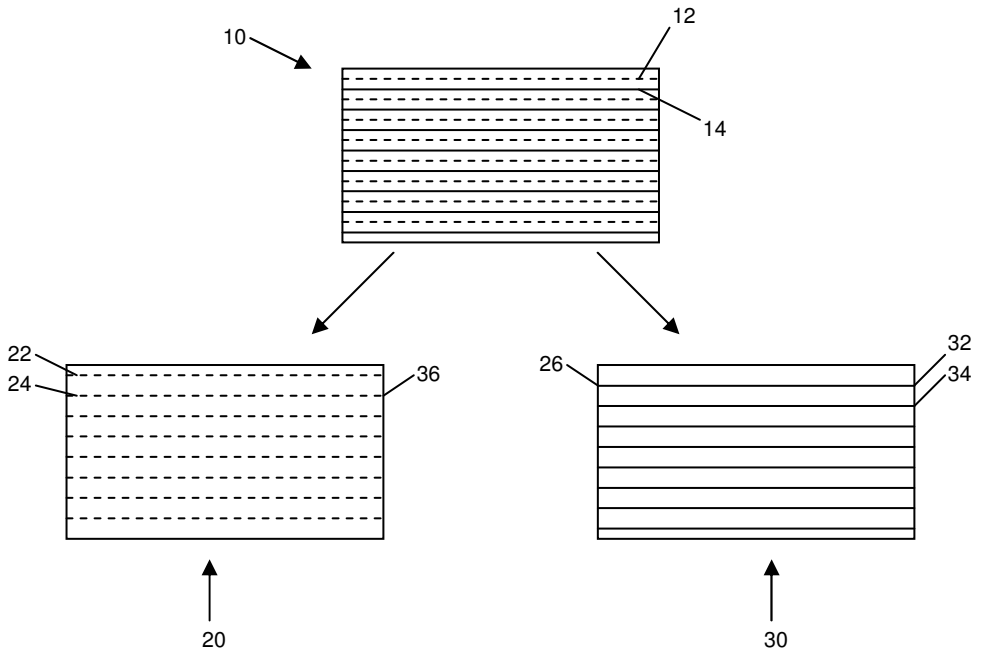


Figure 1

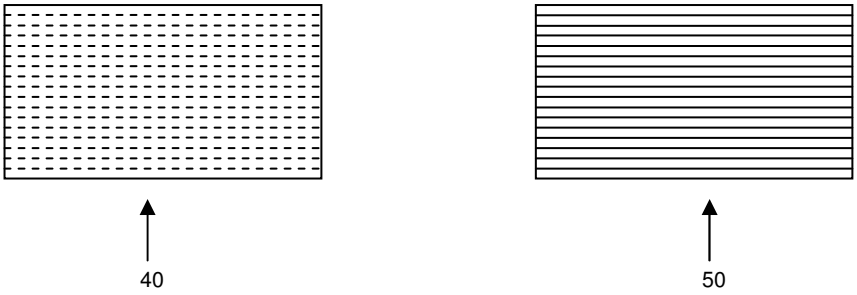


Figure 2

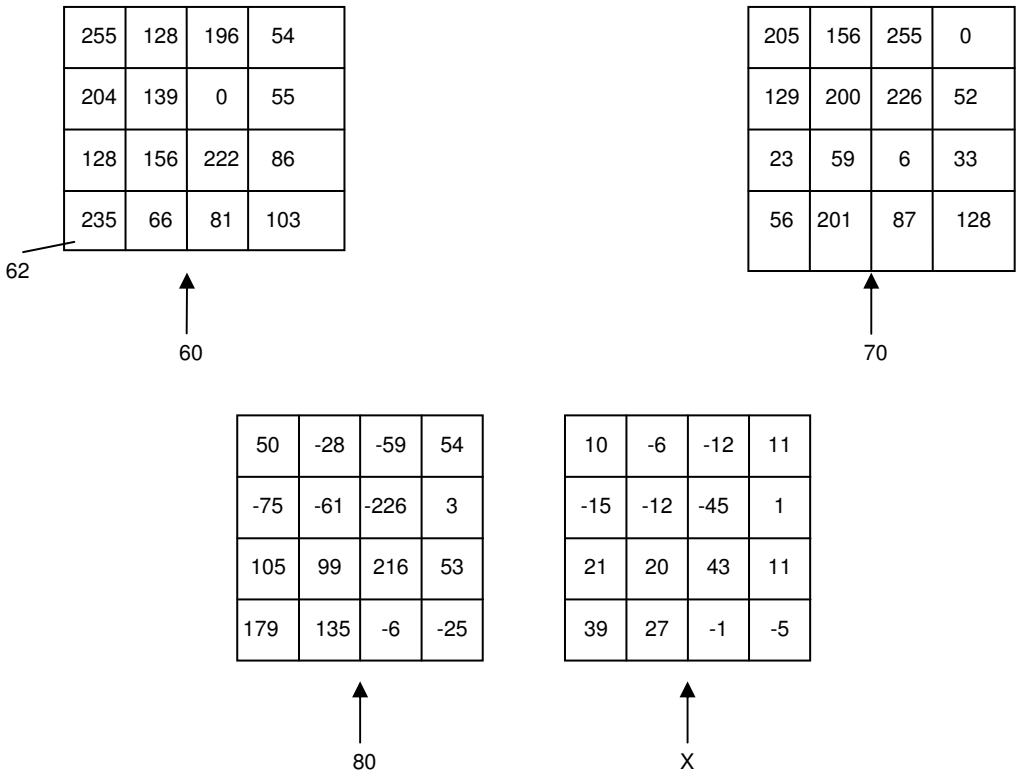

255 128 196 54
204 139 0 55
128 156 222 86
235 66 81 103
62
60
205 156 255 0
129 200 226 52
23 59 6 33
56 201 87 128
70
50 -28 -59 54
-75 -61 -226 3
105 99 216 53
179 135 -6 -25
80
10 -6 -12 11
-15 -12 -45 1
21 20 43 11
39 27 -1 -5
X

Figure 3

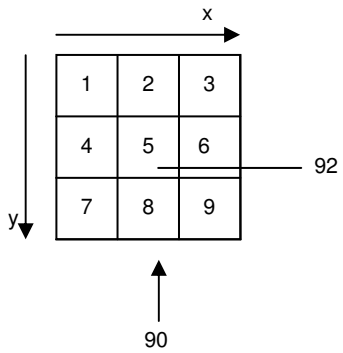


Figure 4

Appendix D

P109986EP/CTW

A METHOD OF DETERMINING FIELD DOMINANCE IN A SEQUENCE OF VIDEO FRAMES

ABSTRACT

A method of determining field dominance in a sequence of video frames, the method comprising: generating from a first video frame a top field and a bottom field; interpolating the top and bottom fields to produce an interpolated top field frame and an interpolated bottom field frame respectively; correlating each of the interpolated top field frame and interpolated bottom field frame with a second video frame occurring immediately previous to the first video frame in the sequence of video frames and with a third video frame occurring immediately subsequent to the first video frame in the sequence of video frames; and determining from the outcome of the correlation the field dominance of the sequence of video frames.

The present invention relates to a method of determining field dominance in a sequence of video frames. Video frames can be classified as either progressive or interlaced, depending upon the method used to display them. In a progressive frame the horizontal lines of pixels that make up the frame are displayed line by line from top to bottom. In contrast, an interlaced frame is created by displaying two fields in turn, one field (known as the top field) containing the top line of the frame and every second subsequent line, and the other field (the bottom field) containing the second line from the top and every subsequent line, thus including the bottom line of the frame. Interlaced frames rely on the fact that it takes time for the first field of displayed pixels to decay from the display apparatus, during which time the second field is displayed, so as to create the illusion of a single frame containing all the lines of pixels.

The fields of an interlaced video frame are captured sequentially, which means that there is a small time delay between the first field to be captured and the second field to be captured. It is possible for the information contained in the scene to change in this time interval and for this reason it is desirable that the fields of the video frame are displayed in the correct order.

Interlaced video frames can be described as either “top field first” or “bottom field first”, depending upon which of the fields making up the frame is intended to be displayed first. As there is small delay between displaying the first field and displaying the second field, the field intended to be displayed second may contain different information from that contained in the field intended to be displayed first, for example if movement has occurred in the frame in the delay between displaying the first and second fields. Such differences between the field intended to be displayed first and the field intended to be displayed second are known as “inter-field motion”. If fields containing inter-field motion are displayed in an incorrect order, distortion may appear in the displayed frame. In an interlaced display, for example, the video typically becomes juddery or shaky as information appears earlier than it was intended to appear. In a progressive display, the reversal of the fields will not cause such juddery or shaky video, as the fields are put together and displayed at a rate of N frames per second, rather than 2N fields per second, but regardless of the field order, the inter-field motion will lead to combing artefacts, i.e. areas of the frames where rows of lines appear, giving a “combed” appearance.

The property of a sequence of video frames by which the sequence can be described as either “top field first” or “bottom field first” is referred to as the field dominance (or field polarity) and is generally dictated by the video standards under which the video sequence is either recorded or intended to be displayed. For example, the most popular European broadcast standard is PAL (phase alternating line) and has top field first field dominance, whereas the American broadcast standard is NTSC (national television systems committee) which has bottom field first field dominance. If a video sequence having a particular field dominance is

played back through a video system configured to play video sequences of the opposite field dominance, or in other words if the field order is reversed, severe visual artefacts may be produced, for example any motion in the video sequence may have a juddering and jittery appearance. Such artefacts will only occur when the video sequence is displayed on an interlaced display but will not be visible when viewed on a progressive display, as in such a display successive fields are combined together to form a frame for displaying.

Although ideally metadata associated with the video stream, which may take the form of a flag encoded in the video stream, will indicate whether a particular video sequence should be top field first or bottom field first, it is possible that either the metadata is not set correctly, possibly as a result of an editing process, or the intended playback equipment, for example video decoder within a digital set top box, is not configured so as to be able to either read the metadata or take the metadata into consideration. It would therefore be beneficial to video producers and broadcasters to be able to quickly and easily determine the field dominance in a video sequence.

According to a first aspect of the present invention there is provided a method of determining field dominance in a sequence of video frames, the method comprising: generating from a first video frame, a top field and a bottom field; interpolating the top and bottom fields to produce an interpolated top field frame and an interpolated bottom field frame respectively; correlating each of the interpolated top field frame and interpolated bottom field frame with a second video frame occurring immediately previous to the first video frame in the sequence of video frames and with a third video frame occurring immediately subsequent to the first video frame in the sequence of video frames; applying a metric to the outcome of the correlation; and determining from the applied metric the field dominance of the sequence of video frames.

The interpolated top field frame may be produced by averaging adjacent lines of the top field and the interpolated bottom field frame may be produced by

averaging adjacent lines of the bottom field. When the correlation to the interpolated top field frame is greater with the previous frame than with the subsequent frame and the correlation to the interpolated bottom field frame is greater with the subsequent frame than the previous frame, then the field dominance is preferably determined to be top field first.

Similarly, when the correlation to the interpolated top field frame is greater with the subsequent frame than with the previous frame and the correlation to the interpolated bottom field is greater with previous frame than with the subsequent frame, then the field dominance is preferably determined to be bottom field first. Prior to performing the determination step the method may further comprise: calculating a first difference value between correlation outcomes of the correlation at the interpolated top field frame to the previous frame and the correlation of the interpolated top field frame to the subsequent frame; calculating a second difference value between the correlation outcomes of the correlation of the interpolated bottom field frame to the previous frame and the correlation of the interpolated bottom field frame to the subsequent frame; and determining the field dominance only if the first and second difference values are greater than a predetermined threshold value.

Additionally, the threshold value is preferably determined by calculating the first and second difference values for a known sequence of static frames. The method may further comprise, subsequent to the correlation step, counting the number of pixels for which the difference in pixel value between a pixel in each of the interpolated top field frame and interpolated bottom field frame and a corresponding pixel in a first reference frame is less than the difference in pixel value between the pixel and a corresponding pixel in a second reference frame, wherein the number of pixels is counted for each of the interpolated top and bottom field frames when the first reference frame comprises the previous frame and the second reference frame comprises the subsequent frame and when the first reference frame comprises the subsequent frame and the second reference frame comprises the previous frame.

According to a further aspect of the present invention there is also provided a computer program for performing the method of the first aspect of the invention.

Embodiments of the present invention will now be described below by way of non-limiting illustrative example only, with reference to the accompanying figures, of which: Figure 1 schematically illustrates an interlaced video frame; Figure 2 schematically illustrates generating a pair of video fields from an interlaced video frame; Figure 3 schematically illustrates a pair of interpolated top and bottom field frames; and Figure 4 schematically illustrates the principle of temporal correlation between individual fields of a frame and previous or subsequent frames.

Referring to Figure 1, a video frame 10 is schematically illustrated that comprises horizontal lines 12, 14 that make up an image. Typically, a frame conforming to the PAL standard comprises 625 such lines of pixels, whilst a frame conforming to the US NTSC standard comprises 525 lines. As previously mentioned, each video frame 10 comprises two separate fields. One field will contain the top line of pixels and every subsequent second line, i.e. it will contain all of the broken lines illustrated in the representation of Figure 1. This field is referred to as the top field. The other field will contain the second line of pixels and every subsequent second line, such that it includes the bottom line of pixels in the video frame, i.e. the solid line of pixels represented in Figure 1. This field is referred to as the bottom field.

Although individual video sequences will be recorded with a constant, single, field dominance, it is quite likely that a number of such individual video sequences will be edited together to form the final broadcast video and it is probable that different individual video sequences will have different field dominance, since the individual video sequences may be captured and collated using the differing broadcast standards available and applicable. As previously noted, if a sequence of video frames are displayed with the field order reversed then severe visual artefacts will tend to be produced when the edited sequence is viewed on an interlaced

display. It is therefore extremely useful and desirable when editing a number of video sequences to be aware of the field dominance of each video sequence to ensure that field dominance is preserved in the final edited video sequence.

To determine the field dominance according to embodiments of the present invention an individual video frame 10 must be divided into top and bottom fields. Referring to Figure 2, the top field 30 is generated by extracting the top line 12 of pixels from the frame 10 and every second subsequent line of pixels and storing these lines in the position from which they were extracted in the frame 10 in the top field 30. Similarly, the bottom field 40 is generated by extracting the second line 14 of pixels and every subsequent second line of pixels and storing them in the position from which they were extracted from the frame 10 in the bottom field 30.

The top and bottom fields 30, 40 each contain only half of the information contained in the video frame 10 from which they were generated. Therefore, the top and bottom fields must be interpolated to produce top and bottom field frames each containing as much information as the video frame 10. Any interpolation method may be used in embodiments of the present invention, however in the embodiment illustrated in Figure 2 adjacent lines of pixels in the field to be interpolated are averaged. Thus, for example, to generate the second line of an interpolated top field frame, as illustrated at 50 in Figure 3, the value of each pixel of the top line 32 of the top field 30 is summed with the value of the corresponding pixel of the second line 34 of the top field 30and divided by 2 to obtain an average pixel value and the "missing" second line of the top field 30 is built up from the average pixel values calculated in this way.

Similarly, to generate the second line of an interpolated bottom field frame, shown as 60 in Figure 3, the value of each pixel of the first line 42 of the bottom field 40 is summed with the value of the corresponding pixel of the second line 44 of the bottom field 40 and resulting sum of pixel values is divided by 2 to obtain an average pixel value and the "missing" second line of the bottom field 40 is built up

from the average pixel values calculated in this way. This process is repeated to generate, from the top and bottom fields 30, 40, interpolated top and bottom field frames 50, 60, each of which contains as much information as the frame 10 from which the top and bottom fields 30, 40 were generated. The interpolated top and bottom field frames 50, 60 are effectively progressive frames which represent the information that can be seen at the time at which each of the top and bottom fields 30, 40 are displayed in an interlaced system.

The interpolated top and bottom field frames 50, 60 are then each correlated with the previous frame in the video sequence to the frame from which the interpolated field frames have been generated and also correlated with the next frame in the video sequence. The rationale for performing this correlation process is derived from the knowledge that the time difference between two frames in a video sequence is inversely proportional to the correlation between them. This principle can also be applied to the separate fields that constitute each frame. The field to be displayed first in a particular frame will have a closer relation to the preceding frame in the video sequence, whilst the field to be displayed second will have a closer correlation to the succeeding frame. A diagrammatic representation of this principle is shown in Figure 4 in which a sequence of video frames 70 is illustrated, with each frame comprising a top and bottom field 20. In the sequence illustrated in Figure 4 the field dominance is top field first. It can be seen that the top field of the Nth frame (and therefore the interpolated top field frame derived from it) has a closer temporal and spatial correlation with the preceding, N-1, frame, whilst the bottom field of the Nth frame has a closer temporal and spatial correlation with the succeeding, N+1, frame.

As previously mentioned, both the interpolated top field frame (X_T) and the interpolated bottom field frame (X_B) are correlated with the previous frame (X_p) and the next future frame (X_f) such that for each frame in the video sequence four separate correlation values are obtained:

a = correlation (X_T, X_p)

b = correlation (X_B, X_f)

c = correlation (X_T, X_f)

d = correlation (X_B, X_p)

Any suitable metric may be used to measure the correlation, such as peak signal to noise ratio (PSNR), mean square error (MSE) or mean absolute error (MAE). In preferred embodiments a correlation difference factor, Δ, is calculated and a check using a correlation difference factor Δ is performed as follows:

Δ = Abs (a-c)

Δ = Abs (b-d)

Δ > threshold

The correlation difference factor Δ indicates the degree of similarity between the frames for which the correlation has to be calculated. A high value of the correlation difference factor indicates that there is low similarity between the frames and therefore there is more active motion information available for processing, thereby improving the reliability of the method. A low value of the correlation difference factor Δ indicates that the frames are similar and may consequently lead to false positives, as there is no significant activity happening between the frames. Consequently, the threshold for the correlation difference factor Δ is preferably determined from the correlation difference factor outcomes for known static frames. If this correlation difference factor is less than the predetermined threshold, then the field order of the frame is considered indeterminate. If the correlation difference factor is greater than the threshold then the following table shows the possible results of the correlation check and their interpretation.

Number	Condition	Interpretation
1	a > c and b > d	Field order = top field first
2	a < c and b < d	Field order = bottom field first
3	other conditions	Indeterminate result

Referring to this table it can be seen that for condition 3 the result achieved is indeterminate, i.e. is not possible to infer from the correlation result what the field dominance is and there is an equal probability of both bottom and top fields being displayed first in order. This indeterminate condition typically happens when one of the neighbouring frames is static or conversely when there is a significant texture change among the frames. In further embodiments of the present invention further processing of the frames is performed to try and resolve those indeterminate results from the frame correlation process. In further embodiments of the present invention the further processing technique applied comprises determining the level of correlation in luminance value pixel by pixel between each of the interpolated field frames and the previous and future frame within the video sequence. This metric determines pixel by pixel if the pixel value of the interpolated field frame is closer to the value of a corresponding pixel in a first reference frame than to the value of the corresponding pixel in a second reference frame. Depending upon the outcome a counter is either incremented or decremented and the final counter value represents the outcome of the correlation calculation. For each field frame two correlation calculations are performed, a first calculation in which the first reference frame is the previous frame in the video sequence and the second reference frame is the future frame, and a second calculation in which the first reference frame is the future frame and the second reference frame is the previous frame. The four possible correlation calculations for each pair of interpolated frames are indicated below, in which the correlation metric is referred to as the optical flow.

Oa = optical flow (X_T, X_p, X_f)

Ob = optical flow (X_T, X_f, X_p)

Oc = optical flow (X_B, X_p, X_f)

Od = optical flow (X_B, X_f, X_p)

The optical flow metric effectively looks at the direction of movement (or optical flow) between successive frames, rather than simply the magnitude of change in pixel luminance value. This reduces the possible masking effect of a large change in luminance value occurring over only a small area of an image frame, which would tend to generate a false positive in the initial correlation calculation as described above. The outcome of the four optical flow correlation determinations are subsequently interpreted as follows:

Number	Condition	Interpretation
1	oa > ob and oc < od	Top field first
2	oa < ob and oc > od	Bottom field first
3	other conditions	Inconclusive result

As can be seen from the condition table above, it is still possible after the further processing that the result of the field order determination is inconclusive, in which case the result for that frame is either disregarded or is recorded as inconclusive

Nonetheless, the above described method for detecting field dominance according to the present invention provides a robust output across video data streams of different bit rate, resolution and quality. A particular advantage of the method of the present invention is that it does not generate false positives for a progressive frame, since the field dominance has no influence on such frames such that the method of the present invention will generate an inconclusive result for such progressive frames.

CLAIMS

1. A method of determining field dominance in a sequence of video frames, the method comprising:

- generating from a first video frame a top field and a bottom field;
- interpolating the top and bottom fields to produce an interpolated top field frame and an interpolated bottom field frame respectively;
- correlating each of the interpolated top field frame and interpolated bottom field frame with a second video frame occurring immediately previous to the first video frame in the sequence of video frames and with a third video frame occurring immediately subsequent to the first video frame in the sequence of video frames; and
- determining from the outcome of the correlation the field dominance of the sequence of video frames.

2. The method of claim 1 wherein the interpolated top field frame is produced by averaging adjacent lines of the top field and the interpolated bottom field frame is produced by averaging adjacent lines of the bottom field

3. The method of claim 1 or 2, wherein when the correlation to the interpolated top field frame is greater with the previous frame than with the subsequent frame and the correlation to the interpolated bottom field frame is greater with the subsequent frame than the previous frame, then the field dominance is determined to be top field first.

4. The method of any preceding claim, wherein when the correlation to the interpolated top field frame is greater with the subsequent frame than with the previous frame and the correlation to the interpolated bottom field is greater with previous frame than with the subsequent frame, then the field dominance is determined to be bottom field first.

5. The method of any preceding claim, wherein prior to performing the determination step the method further comprises:

- calculating a first difference value between correlation outcomes of the correlation of the interpolated top field frame to the previous frame and the correlation of the interpolated top field frame to the subsequent frame;
- calculating a second difference value between the correlation outcomes of the correlation of the interpolated bottom field frame to the previous frame and the correlation of the interpolated bottom field frame to the subsequent frame; and
- determining the field dominance only if the first and second difference values are greater than a predetermined threshold value.

6. The method of claim 5, wherein the threshold value is determined by calculating the first and second difference values for a known sequence of static frames.

7. The method of any preceding claim further comprising, subsequent to the correlation step, counting the number of pixels for which the difference in pixel value between a pixel in each of the interpolated top field frame and interpolated bottom field frame and a corresponding pixel in a first reference frame is less than the difference in pixel value between the pixel and a corresponding pixel in a second reference frame, wherein the number of pixels is counted for each of the interpolated top and bottom field frames when the first reference frame comprises the previous frame and the second reference frame comprises the subsequent frame and when the first reference frame comprises the subsequent frame and the second reference frame comprises the previous frame.

8. A computer program for performing the method of any one of claims 1 to 7.

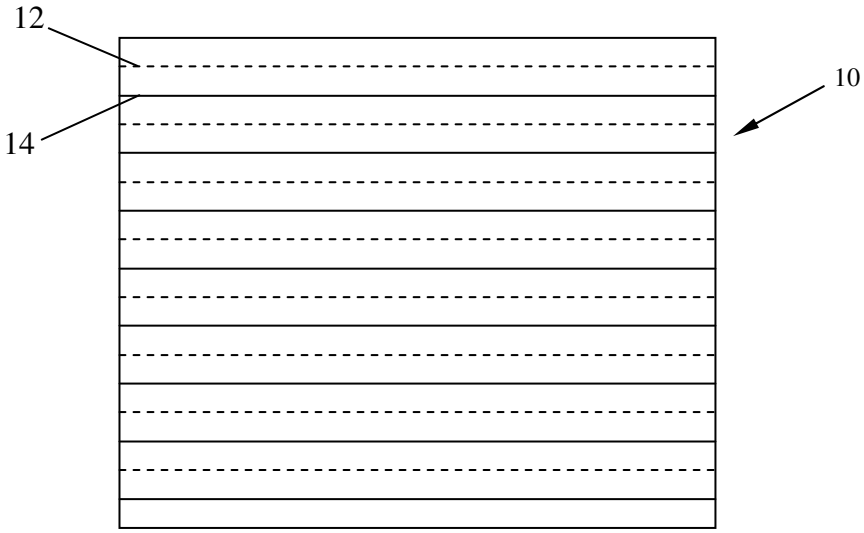


Figure 1

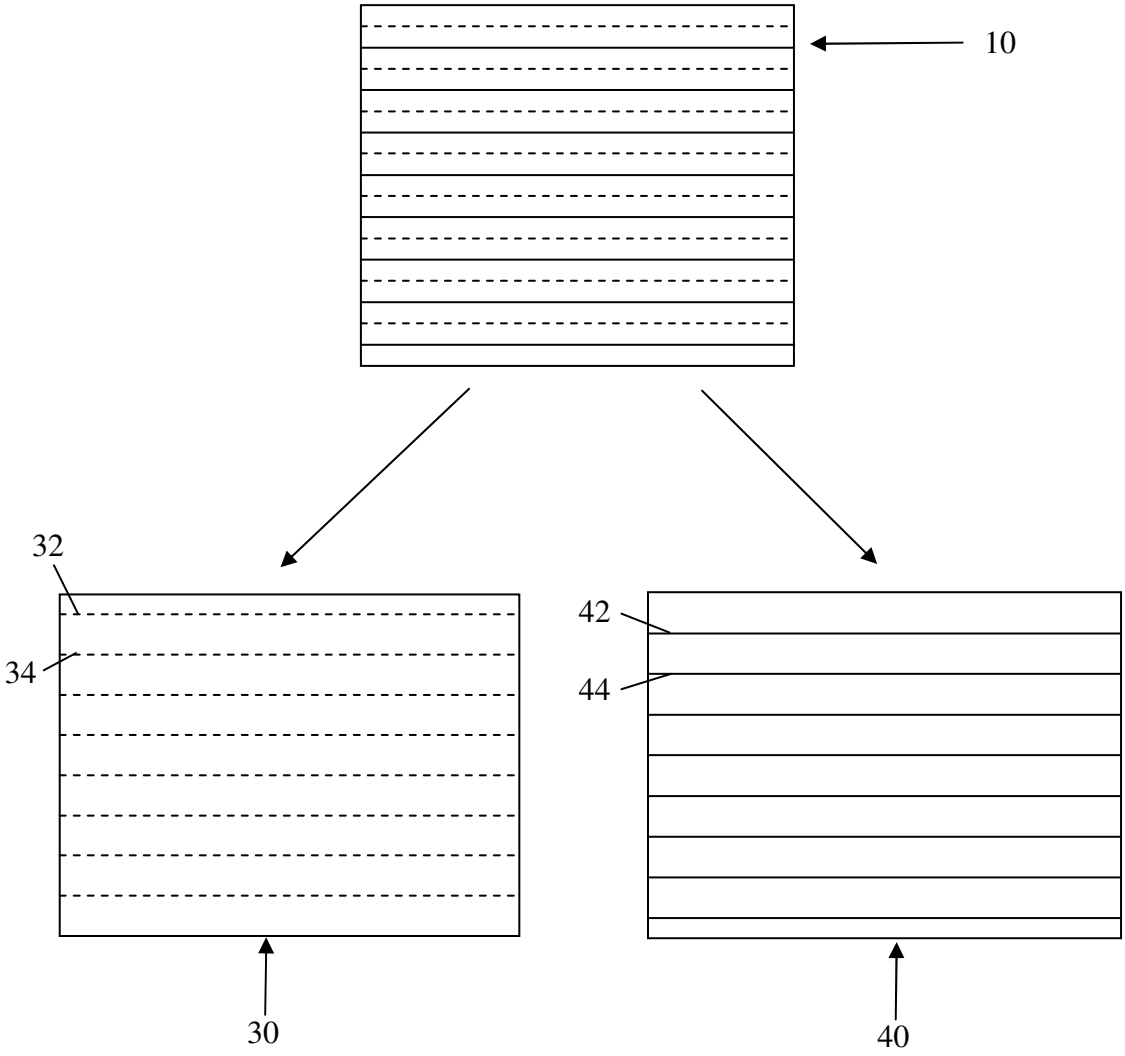


Figure 2

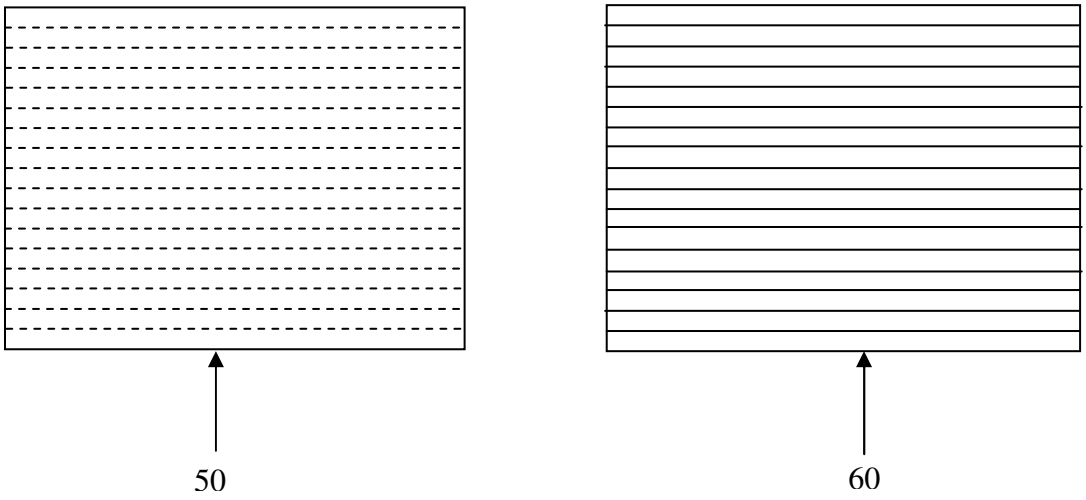


Figure 3

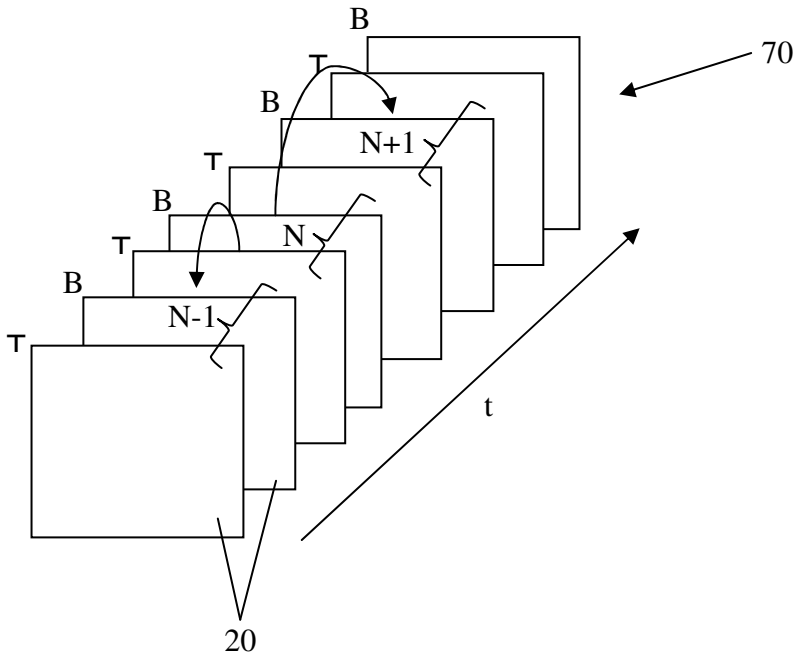


Figure 4

Appendix E

P109983EP/MH

A METHOD OF TRACKING A REGION OF INTEREST IN A SEQUENCE OF VIDEO FRAMES

ABSTRACT

A method of tracking a region of interest in a sequence of video frames, the method comprising performing a search to locate a region of interest (52) in a first frame (50) of the sequence and performing a search to locate a corresponding region of interest in a subsequent frame (60) of the sequence, characterised in that the search in the subsequent frame (60) is commenced in an area of the subsequent frame (60) which relates to an area in the first frame (50) in which a region of interest (52) was identified by the search of the first frame (50).

The present invention relates to a method of tracking a region of interest in a sequence of video frames. In the field of video processing, there are many applications which require a video frame, or a sequence of video frames, to be processed to determine or assess a property of the frame or sequence of frames. For example, a video frame may be classified as interlaced or progressive on the basis of such a processing operation. In many such operations processing of the entire frame is not necessary as a result (for example an interlaced/progressive classification) can be obtained by processing only part of the frame containing relevant information. This is known as Region of Interest (ROI) processing, and is used to reduce the processing time required to obtain a result for the frame being processed. The region of interest may be related to the content of the video frame, or it may relate to an artefact of the frame such as inter-field motion.

Typically, a region of interest is identified by processing blocks of pixels of the frame in turn until one or more of the blocks is deemed to meet some criteria

indicating that it comprises or is part of a region of interest. The blocks may be, for example, square blocks containing 4096 pixels (i.e. 64 pixels by 64 pixels). A spiral search method is often used to identify a region of interest, whereby a block in or towards the centre of the frame is processed initially, followed by further blocks in an increasing spiral, as is shown in Figure 1.

If the region of interest is to be tracked in subsequent frames in a sequence of video frames, the spiral search method is typically used to locate the region of interest in each subsequent frame, and this can lead to unnecessarily long processing times as the region of interest is identified.

According to a first aspect of the invention, there is provided a method of tracking a region of interest in a sequence of video frames, the method comprising performing a search to locate a region of interest in a first frame of the sequence and performing a search to locate a corresponding region of interest in a subsequent frame of the sequence, characterised in that the search in the subsequent frame is commenced in an area of the subsequent frame which relates to an area in the first frame in which a region of interest was identified by the search of the first frame. The search of the subsequent frame may be commenced at a position corresponding to a position at which a region of interest was identified by the search of the first frame. The search of the first frame may terminate when a region of interest is identified. The search of the first frame and the search of the subsequent frame may comprise spiral searches. The search of the subsequent frame may commence in a direction corresponding to a direction of the search of the first frame when the search of the first frame terminated. The searches of the first frame and of the subsequent frame are preferably performed in relation to blocks of pixels of each of the first and subsequent frames. According to a second aspect of the invention, there is provided a computer program for performing the method of the first aspect.

Embodiments of the invention will now be described, strictly by way of example only, with reference to the accompanying drawings, of which Figure 1 is a

schematic illustration of a known spiral search method; Figure 2 is a schematic illustration showing a spiral search used to locate a region of interest in a first video frame of a video sequence; Figure 3 is a schematic illustration showing a spiral search used to locate the region of interest of Figure 2 in a subsequent frame using the method of the invention; and Figure 4 is a schematic illustration showing a spiral search used to locate the region of interest of Figure 2 in a subsequent frame using an alternative embodiment of the method of the invention. Referring first to Figure 1, there is shown a schematic representation of a known spiral search method, which may be used to identify a region of interest such as inter-field motion in a video frame. In Figure 1 a video frame is shown at 10 and can be thought of as being made up of blocks 12, each of the blocks comprising a number of pixels. In the known spiral search method the search typically commences at a block 14 in or towards the centre of the video frame because the region of interest is more likely to be located in a central portion of the video frame 10 than in an outer portion. Thus, the central block 14 of pixels is processed using appropriate algorithms to determine whether it contains a region of interest. If no region of interest is found in the central block an adjacent block 16 is processed to determine whether it contains a region of interest. In the example shown in Figure 1, the block 16 is located to the right of the central block 14, but it will be appreciated that any adjacent block may be selected for processing.

If no region of interest is found in the block 16 the search changes direction and a further block 18, which in this example is located above the block 16, is processed. If no region of interest is found in the block 18 the search moves to an adjacent block 20, which is processed to determine whether it contains a region of interest. The blocks of the video frame 10 are processed in turn, with the search method following an outwardly extending spiral path through the blocks until a region of interest is identified, at which point the search may stop. Alternatively, the position of the region of interest may be recorded before the search recommences in order to identify further regions of interest in the video frame 10.

In the event that the spiral path followed by the search reaches an edge of the video frame 10, as occurs at blocks 22, 24, 26, 28, the search recommences in the next unsearched block, as if the spiral path had not been broken, as indicated by dashed lines 40, 42 and 44 in Figure 1. Thus, when block 2 has been processed, the search moves onto block 30, whilst processing of block 24 is followed by processing of block 36. This spiral search method is an efficient way of processing a single video frame to identify one or more regions of interest. However, where a sequence of video frames must be processed, to identify a region of interest in a first frame and track that region of interest in one or more subsequent frames, performing a spiral search commencing at a central block of the or each subsequent frame to identify the region of interest can take an unnecessarily long time.

In most cases where a region of interest occurs in a sequence of video frames, the regions of interest between frames are spatially correlated, that is to say the region of interest in a subsequent frame is located in the same or a similar area of the frame as the region of interest in a first frame. Successive frames in a sequence of video frames generally do not differ by a large amount, and the displacement of regions of interest between successive frames in a sequence is usually small. The present invention makes use of these properties to reduce the time taken to identify a region of interest in a subsequent frame of a video sequence, thus reducing the processing resources required to process the sequence and the power consumption of a device on which the method runs. In the method of the present invention a search of a subsequent frame commences at a position in an area of the subsequent frame which relates to an area in the previous frame in which a region of interest was identified, as will now be described with reference to Figures 2 and 3.

In Figure 2 a spiral search performed on a first video frame 50 in a sequence has identified a region of interest 52 in a block 54 of the video frame 50. The position of the region of interest 52 is recorded for use in determining a position at which to commence a search of a subsequent frame of the sequence. Figure 3 shows such a subsequent frame 60, in which the region of interest 52 has moved to a block

62. A spiral search of the subsequent frame 60 commences at a point 64, which corresponds to the position of the region of interest 52 in the first frame 50. As the path of the spiral search when the region of interest 52 in the first frame 50 was identified was following an upward direction, the spiral search of the subsequent frame 60 preferably commences in an upward direction, as this is the likely direction of movement of the region of interest 52. If the path of the spiral search when the region of interest 52 in the first frame 50 was identified was following a different direction, the spiral search of the subsequent frame 60 would preferably commence in that different direction. The region of interest 52 is identified in the block 62, and its position is recorded for use in determining a position at which to commence a search of a subsequent frame in the sequence.

Alternatively, the spiral search of the subsequent frame 60 need not necessarily commence in the direction which was being followed by the spiral search of the first frame 50 when the region of interest 52 was identified. For example to simplify execution of the method the spiral search of the subsequent frame 60 may always commence in the same direction, for example upwards. Figure 4 illustrates an alternative embodiment of the method, in which the spiral search of the subsequent frame 60 does not commence at a point in the subsequent frame 60 corresponding to the exact position at which the region of interest 52 was identified in the first frame 50, but commences in an area of the subsequent frame 60 which relates to the area in the first frame 50 in which the region of interest 52 was identified. In this example, a point corresponding to a position slightly further backwards on the spiral search path of the first frame 50 at which the region of interest 52 was identified is selected as the point 66 at which the search of the subsequent frame 60 commences. The reason for selecting the point 66 as the point at which to commence the search of the subsequent frame 60 is that it reduces the likelihood that an artefact occurring in the first frame 50 incorrectly identified as a region of interest by the search of the first frame 50 will “misdirect” the search of the second frame 60, as the search of the subsequent frame has to follow a longer spiral path than in the embodiment of Figure 3, processing more blocks of the subsequent frame 60, thus increasing the likelihood

of correctly identifying a region of interest before identifying the incorrectly identified artefact of the first frame 50. Of course, the improved likelihood of correctly identifying a region of interest in the subsequent frame 60 comes at the expense of slightly increased processing time. However, this processing time is still reduced in comparison to the processing time required to perform a new spiral search of the subsequent frame 60 commencing at a central block of the subsequent frame 60.

It will be appreciated that the point 66 selected as the point at which the spiral search of the subsequent frame 60 commences may be any point in the subsequent frame 60 which is in an area of the subsequent frame 60 which relates to an area of the first frame 50 in which the region of interest 52 was identified. For example, the point 66 may correspond to any position on the spiral search path of the first frame 50 on which the region of interest was identified, except the position at which the spiral search of the first frame 50 commenced.

In either embodiment of the method, if the region of interest 52 identified by the search of the first frame 50 is located at a position in the first frame higher than the position at which the search of the first frame 50 commenced, i.e. if the region of interest 52 in the first frame 50 is located above the central block 14 of the first frame 50, it is likely that the region of interest 52 will have moved in an upward direction in the subsequent frame 60, and thus the search of the subsequent frame 60 may commence at a point slightly upwards of a point corresponding to the position of the region of interest 52 in the first frame 50, to reduce further the number of blocks of the subsequent frame 60 that must be processed, and thus the time taken, to identify the region of interest 52 in the subsequent frame 60. Similarly, if the region of interest 52 identified by the search of the first frame 50 is located at a position in the first frame lower than the position at which the search of the first frame 50 commenced, i.e. if the region of interest in the first frame 50 is located below the central block 14 of the first frame 50, it is likely that the region of interest 52 will have moved in a downward direction in the subsequent frame, and thus the search of

the subsequent frame 60 may commence at a point slightly downwards of a point corresponding to the position of the region of interest 52 in the first frame 50, to reduce further the number of blocks in the subsequent frame 60 that must be processed.

The algorithm or method used to process the blocks of the first and subsequent frames 50, 60 will depend upon the type of region of interest to be identified, as will be apparent to those skilled in the art. For example, if inter-field motion is to be identified, a correlation between top and bottom fields of the frame may be performed. Although the method of the present invention has been described in relation to a spiral search, it will be appreciated that it can also be used in conjunction with other search methods to reduce the processing time required to identify and track a region of interest in a sequence of video frames.

CLAIMS

1. A method of tracking a region of interest in a sequence of video frames, the method comprising performing a search to locate a region of interest in a first frame of the sequence and performing a search to locate a corresponding region of interest in a subsequent frame of the sequence, characterised in that the search in the subsequent frame is commenced in an area of the subsequent frame which relates to an area in the first frame in which a region of interest was identified by the search of the first frame.

2. A method according to claim 1 wherein the search of the subsequent frame is commenced at a position corresponding to a position at which a region of interest was identified by the search of the first frame.

3. A method according to claim 1 or claim 2 wherein the search of the first frame terminates when a region of interest is identified.

4. A method according to any one of the preceding claims wherein the search of the first frame and the search of the subsequent frame comprise spiral searches.

5. A method according to any one of the preceding claims wherein the search of the subsequent frame commences in a direction corresponding to a direction of the search of the first frame when the search of the first frame terminated.

6. A method according to any one of the preceding claims wherein the searches of the first frame and of the subsequent frame are performed in relation to blocks of pixels of each of the first and subsequent frames.

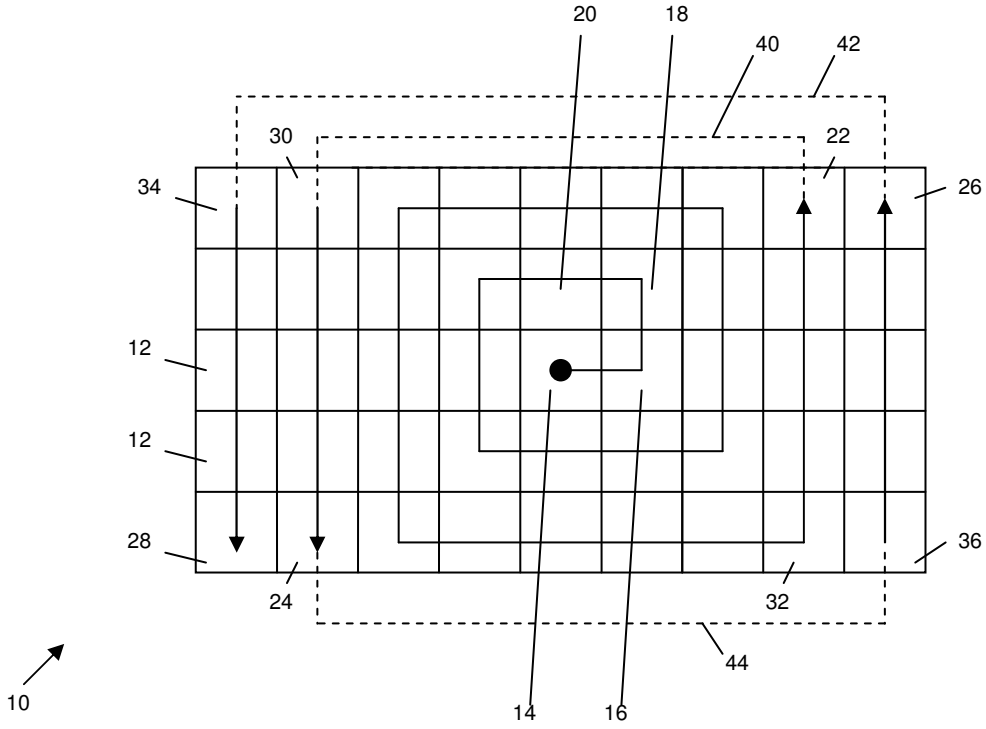


Figure 1

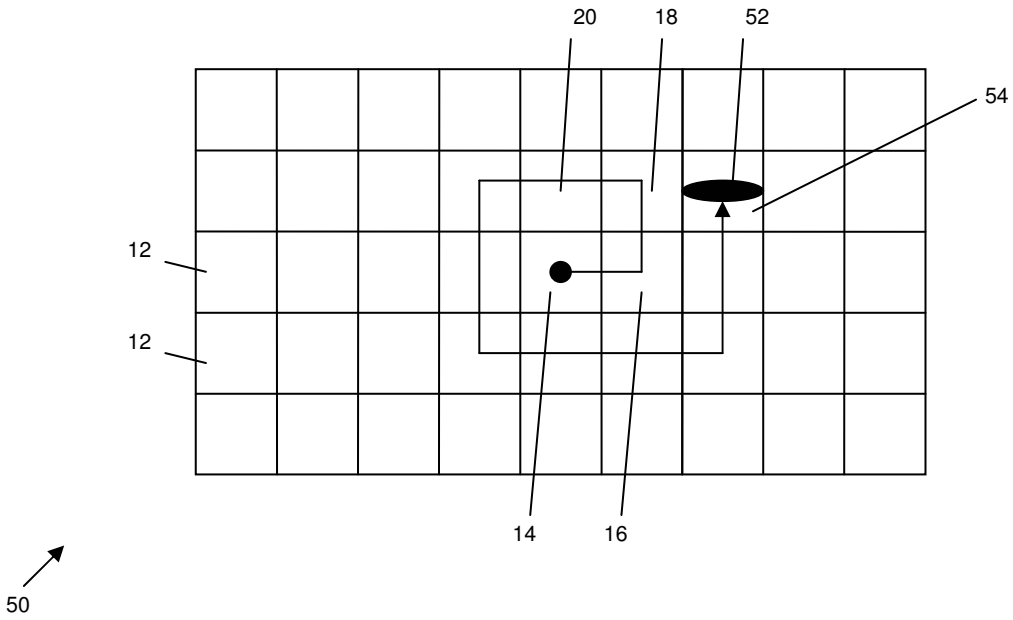


Figure 2

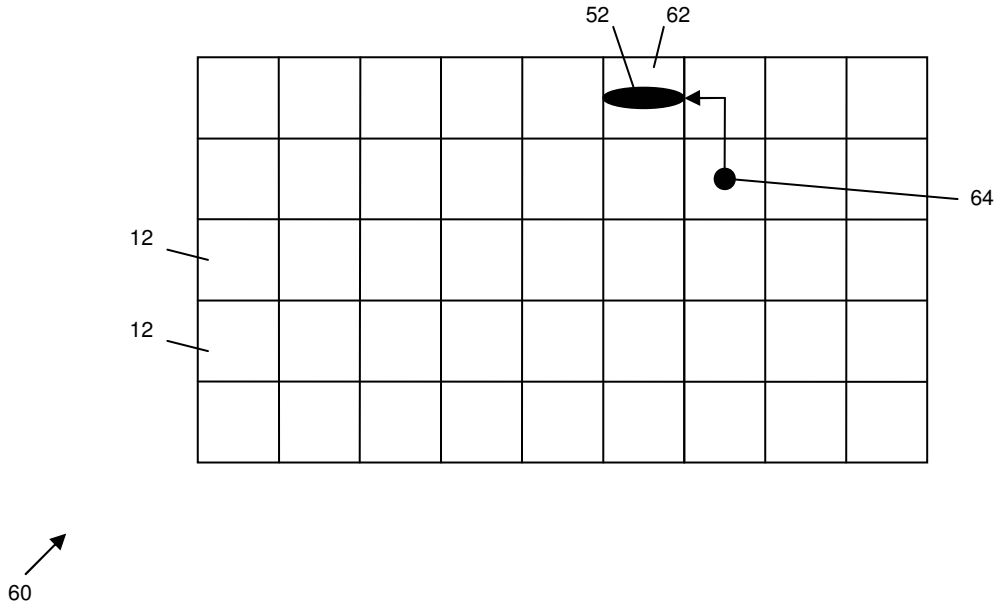


Figure 3

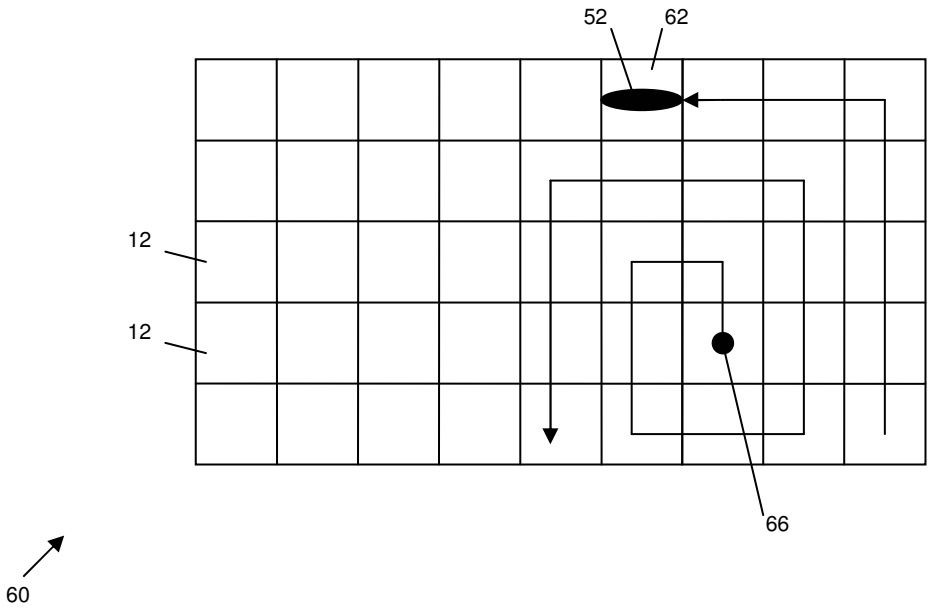


Figure 4

Appendix F

P109982EP/CTW

A METHOD OF IDENTIFYING INCONSISTENT FIELD DOMINANCE METADATA IN A SEQUENCE OF VIDEO FRAMES

ABSTRACT

Embodiments of the present invention provide a method of identifying inconsistent field order flags for a sequence of video frames comprising: for each frame in the sequence of video frames analysing the frame to make an initial determination of the field order for that frame; averaging the initial determination of the field order over a predefined number of most recently analysed frames; and determining those frames for which the averaged field order does not match the field order identified by a respective field order metadata item associated with each frame by comparing the averaged field order for each frame to the respective field order metadata item.

Video frames can be classified as either progressive or interlaced, depending upon the method used to display them. In a progressive frame the horizontal lines of pixels that make up the frame are displayed line by line from top to bottom. In contrast, an interlaced frame is created by displaying two fields in turn, one field (known as the top field) containing the top line of the frame and every second subsequent line, and the other field (the bottom field) containing the second line from the top and every subsequent line, thus including the bottom line of the frame. Interlaced frames rely on the fact that it takes time for the first field of displayed pixels to decay from the display apparatus, during which time the second field is displayed, so as to create the illusion of a single frame containing all the lines of pixels.

The fields of an interlaced video frame are captured sequentially, which means that there is a small time delay between the first field to be captured and the second field to be captured. It is possible for the information contained in the scene to change in this time interval and for this reason it is desirable that the fields of the video frame are displayed in the correct order.

Interlaced video frames can be described as either “top field first” or “bottom field first”, depending upon which of the fields making up the frame is intended to be displayed first. As there is small delay between displaying the first field and displaying the second field, the field intended to be displayed second may contain different information from that contained in the field intended to be displayed first, for example if movement has occurred in the frame in the delay between displaying the first and second fields. Such differences between the field intended to be displayed first and the field intended to be displayed second are known as “inter-field motion”. If fields containing inter-field motion are displayed in an incorrect order, distortion may appear in the displayed frame. In an interlaced display, for example, the video typically becomes juddery or shaky as information appears earlier than it was intended to appear. In a progressive display, the reversal of the fields will not cause such juddery or shaky video, as the fields are put together and displayed at a rate of N frames per second, rather than 2N fields per second, but regardless of the field order, the inter-field motion will lead to combing artefacts, i.e. areas of the frames where rows of lines appear, giving a “combed” appearance.

The property of a sequence of video frames by which the sequence can be described as either “top field first” or “bottom field first” is referred to as the field dominance (or field polarity) and is generally dictated by the video standards under which the video sequence is either recorded or intended to be displayed. For example, the most popular European broadcast standard is PAL (phase alternating line) and has top field first field dominance, whereas the American broadcast standard is NTSC (national television systems committee) which has bottom field

first field dominance. If a video sequence having a particular field dominance is played back through a video system configured to play video sequences of the opposite field dominance, or in other words if the field order is reversed, severe visual artefacts may be produced, for example any motion in the video sequence may have a juddering and jittery appearance. Such artefacts will only occur when the video sequence is displayed on an interlaced display but will not be visible when viewed on a progressive display, as in such a display successive fields are combined together to form a frame for displaying.

Metadata contained within the video stream will typically include a flag indicating whether a particular video frame is encoded as either top field first or bottom field first. However, it is possible for this flag to be either corrupted (or omitted) such that the flag is incorrectly set during video processing, for example as a result of an editing or transcoding action. It would therefore be beneficial to video producers and broadcasters to be able to quickly and easily determine those frames within a video sequence for which the filed dominance flag might be incorrectly set.

The averaging step may comprise allocating as the averaged field order the field order of a predetermined proportion of the predefined number of frames having the same field order, the predetermined proportion of frames preferably comprising at least 30%. Preferably, the predefined number of frames over which the averaging step is performed is 25. According to a further aspect of the present invention there is also provided a computer program arranged to cause a computer to perform the method of the first aspect. Embodiments of the present invention will now be described below by way of non-limiting illustrative example only, with reference to the accompanying figures, of which: Figure 1 schematically illustrates an interlaced video frame; Figure 2 schematically illustrates the editing of two sequences of video fields with opposite field dominance where a field order error may occur if the metadata is constant; Figure 3 schematically illustrates the editing of the two sequences of video fields shown in Figure 2 where a field order error may occur if the metadata is not constant; Figure 4 schematically illustrates generating a pair of

video fields from an interlaced video frame; Figure 5 schematically illustrates a pair of interpolated top and bottom field frames; Figure 6 schematically illustrates the field order averaging process of an embodiment of the present invention; and Figure 7 schematically illustrates the method steps of an embodiment of the present inventions.

Referring to Figure 1, a video frame 10 is schematically illustrated that comprises horizontal lines 12, 14 that make up an image. Typically, a frame conforming to the PAL standard comprises 625 such lines of pixels, whilst a frame conforming to the US NTSC standard comprises 525 lines. As previously mentioned, each video frame 10 comprises two separate fields. One field will contain the top line of pixels and every subsequent second line, i.e. it will contain all of the broken lines illustrated in the representation of Figure 1. This field is referred to as the top field. The other field will contain the second line of pixels and every subsequent second line, such that it includes the bottom line of pixels in the video frame, i.e. the solid line of pixels represented in Figure 1. This field is referred to as the bottom field.

Although individual video sequences will be recorded with a constant, single, field dominance, it is quite likely that a number of such individual video sequences will be edited together to form the final broadcast video and it is probable that different individual video sequences will have different field dominance, since the individual video sequences may be captured and collated using the differing broadcast standards available and applicable. As previously noted, the metadata indicating the field dominance for individual frames or sequences of frames may not be preserved during this editing process or subsequent transcoding processes.

An example of a first editing scenario of a pair of video sequences is schematically illustrated in Figure 2. A first sequence S1 of individual video fields is illustrated with each field 16 is labelled as either a top field T or a bottom field B. In the first field sequence S1 the field dominance is a top field first. A second sequence

of video fields S2 is also illustrated, the field dominance for the second sequence being bottom field first. If the flag in the meta data for the edited sequence 17 is set to top field first for the first 2 frames/4 fields (S1) and then changes to bottom field first for the subsequent 2 frames/4 fields (S2), then there will not be any field dominance errors. However, if the meta data flag points to top field first throughout the edited sequence 17 that would result in video being juddery starting from the 3rd frame/5th field.

An example of second editing scenario of a pair of video sequences is schematically illustrated in Figure 3. In an analogous fashion to Figure 2, first and second sequences S1, S2 of video fields 16 are illustrated, the first sequence S1 being top field first, whilst the second sequence S2 is bottom field first, together with the edited sequence 18. However, in figure 2, the edited sequence 17 is edited such that the second sequence S2 starts with a bottom field , whilst in figure 3, the edit of S2 starts with a top field. If the flag in the meta data for the edited sequence 18 in Figure 3 is set to top field first throughout the sequence starting from the 1st frame, there will not be any field dominance errors. However, if the meta data flag is set to bottom field first from the 3rd frame/5th field of the edited sequence 18, that would result in video being juddery starting from 3rd frame/5th field of the edited sequence 18.

Consequently, according to embodiments of the present invention a consistency check is made between the field order of the video frames indicated by the metadata and the field order determined by analysis of the video frames. The field order may be determined by performing any suitable video analysis technique, such as spatial correlation within and around one or more edges of one or more objects in a frame as disclosed in US patent application US 2006/0139491 A1. However, in preferred embodiments of the present invention the field dominance is determined according to the following method, which is also disclosed in the applicant's co-pending European patent application no. (attorney ref. P109986EP).

To determine the field dominance according to an embodiment of the present invention an individual video frame 10 must be divided into top and bottom fields. Referring to Figure 4, the top field 20 is generated by extracting the top line 12 of pixels from the frame 10 and every second subsequent line of pixels and storing these lines in the position from which they were extracted in the frame 10 in the top field 30. Similarly, the bottom field 30 is generated by extracting the second line 14 of pixels and every subsequent second line of pixels and storing them in the position from which they were extracted from the frame 10 in the bottom field 30.

The top and bottom fields 20, 30 each contain only half of the information contained in the video frame 10 from which they were generated. Therefore, the top and bottom fields must be interpolated to produce top and bottom field frames each containing as much information as the video frame 10. Any interpolation method may be used in embodiments of the present invention, however in the embodiment illustrated in Figure 4 adjacent lines of pixels in the field to be interpolated are averaged. Thus, for example, to generate the second line of an interpolated top field frame, as illustrated at 40 in Figure 5, the value of each pixel of the top line 22 of the top field 20 is summed with the value of the corresponding pixel of the second line 24 of the top field 20. The resulting sum of pixel values is divided by 2 to obtain an average pixel value and the “missing” second line of the top field 20 is built up from the average pixel values calculated in this way.

According to a first aspect of the present invention there is provided a method of identifying inconsistent field order flags for a sequence of video frames comprising: for each frame in the sequence of video frames analysing the frame to make an initial determination of the field order for that frame; averaging the initial determination of the field order over a predefined number of most recently analysed frames; and determining those frames for which the averaged field order does not match the field order identified by a respective field order flag associated with each frame by comparing the averaged field order for each frame to the respective field order flag.

The initial determination of the field order may be indeterminate. Furthermore, the indeterminate field order of an analysed frame may be replaced by the averaged field order. The averaging step may comprise allocating as the averaged field order the field order of a predetermined proportion of the predefined number of frames having the same field order, the predetermined proportion of frames preferably comprising at least 30%. Preferably, the predefined number of frames over which the averaging step is performed is 25. According to a further aspect of the present invention there is also provided a computer program arranged to cause a computer to perform the method of the first aspect. Embodiments of the present invention will now be described below by way of non-limiting illustrative example only, with reference to the accompanying figures, of which: Figure 1 schematically illustrates an interlaced video frame; Figure 2 schematically illustrates the editing of two sequences of video fields with opposite field dominance where a field order error may occur if the metadata is constant; Figure 3 schematically illustrates the editing of the two sequences of video fields shown in Figure 2 where a field order error may occur if the metadata is not constant; Figure 4 schematically illustrates generating a pair of video fields from an interlaced video frame; Figure 5 schematically illustrates a pair of interpolated top and bottom field frames; Figure 6 schematically illustrates the field order averaging process of an embodiment of the present invention; and Figure 7 schematically illustrates the method steps of an embodiment of the present inventions.

Referring to Figure 1, a video frame 10 is schematically illustrated that comprises horizontal lines 12, 14 that make up an image. Typically, a frame conforming to the PAL standard comprises 625 such lines of pixels, whilst a frame conforming to the US NTSC standard comprises 525 lines. As previously mentioned, each video frame 10 comprises two separate fields. One field will contain the top line of pixels and every subsequent second line, i.e. it will contain all of the broken lines illustrated in the representation of Figure 1. This field is referred to as the top field. The other field will contain the second line of pixels and every

subsequent second line, such that it includes the bottom line of pixels in the video frame, i.e. the solid line of pixels represented in Figure 1. This field is referred to as the bottom field.

Although individual video sequences will be recorded with a constant, single, field dominance, it is quite likely that a number of such individual video sequences will be edited together to form the final broadcast video and it is probable that different individual video sequences will have different field dominance, since the individual video sequences may be captured and collated using the differing broadcast standards available and applicable. As previously noted, the metadata indicating the field dominance for individual frames or sequences of frames may not be preserved during this editing process or subsequent transcoding processes.

An example of a first editing scenario of a pair of video sequences is schematically illustrated in Figure 2. A first sequence S1 of individual video fields is illustrated with each field 16 is labelled as either a top field T or a bottom field B. In the first field sequence S1 the field dominance is a top field first. A second sequence of video fields S2 is also illustrated, the field dominance for the second sequence being bottom field first. If the flag in the meta data for the edited sequence 17 is set to top field first for the first 2 frames/4 fields (S1) and then changes to bottom field first for the subsequent 2 frames/4 fields (S2), then there will not be any field dominance errors. However, if the meta data flag points to top field first throughout the edited sequence 17 that would result in video being juddery starting from the 3rd frame/5th field.

An example of second editing scenario of a pair of video sequences is schematically illustrated in Figure 3. In an analogous fashion to Figure 2, first and second sequences S1, S2 of video fields 16 are illustrated, the first sequence S1 being top field first, whilst the second sequence S2 is bottom field first, together with the edited sequence 18. However, in figure 2, the edited sequence 17 is edited such that the second sequence S2 starts with a bottom field , whilst in figure 3, the edit of S2

starts with a top field. If the flag in the meta data for the edited sequence 18 in Figure 3 is set to top field first throughout the sequence starting from the 1st frame, there will not be any field dominance errors. However, if the meta data flag is set to bottom field first from the 3rd frame/5th field of the edited sequence 18, that would result in video being juddery starting from 3rd frame/5th field of the edited sequence 18.

Consequently, according to embodiments of the present invention a consistency check is made between the field order of the video frames indicated by the metadata and the field order determined by analysis of the video frames. The field order may be determined by performing any suitable video analysis technique, such as spatial correlation within and around one or more edges of one or more objects in a frame as disclosed in US patent application US 2006/0139491 A1. However, in preferred embodiments of the present invention the field dominance is determined according to the following method, which is also disclosed in the applicant's co-pending European patent application no. (attorney ref. P109986EP).

To determine the field dominance according to an embodiment of the present invention an individual video frame 10 must be divided into top and bottom fields. Referring to Figure 4, the top field 20 is generated by extracting the top line 12 of pixels from the frame 10 and every second subsequent line of pixels and storing these lines in the position from which they were extracted in the frame 10 in the top field 30. Similarly, the bottom field 30 is generated by extracting the second line 14 of pixels and every subsequent second line of pixels and storing them in the position from which they were extracted from the frame 10 in the bottom field 30.

The top and bottom fields 20, 30 each contain only half of the information contained in the video frame 10 from which they were generated. Therefore, the top and bottom fields must be interpolated to produce top and bottom field frames each containing as much information as the video frame 10. Any interpolation method may be used in embodiments of the present invention, however in the embodiment

illustrated in Figure 4 adjacent lines of pixels in the field to be interpolated are averaged. Thus, for example, to generate the second line of an interpolated top field frame, as illustrated at 40 in Figure 5, the value of each pixel of the top line 22 of the top field 20 is summed with the value of the corresponding pixel of the second line 24 of the top field 20. The resulting sum of pixel values is divided by 2 to obtain an average pixel value and the "missing" second line of the top field 20 is built up from the average pixel values calculated in this way.

Similarly, to generate the second line of an interpolated bottom field frame, shown as 50 in Figure 5, the value of each pixel of the first line 32 of the bottom field 30 is summed with the value of the corresponding pixel of the second line 34 of the bottom field 30. The resulting sum of pixel values is divided by 2 to obtain an average pixel value and the "missing" second line of the bottom field 30 is built up from the average pixel values calculated in this way. This process is repeated to generate, from the top and bottom fields 20, 30, interpolated top and bottom field frames 40, 50, each of which contains as much information as the frame 10 from which the top and bottom fields 20, 30 were generated. The interpolated top and bottom field frames 40, 50 are effectively progressive frames which represent the information that can be seen at the time at which each of the top and bottom fields 30, 40 are displayed in an interlaced system.

The interpolated top and bottom field frames 40, 50 are then each correlated with the previous frame in the video sequence to the frame from which the interpolated field frames have been generated and also correlated with the next frame in the video sequence. The rationale for performing this correlation process is derived from the knowledge that the time difference between two frames in a video sequence is inversely proportional to the correlation between them. This principle can also be applied to the separate fields that constitute each frame. The field to be displayed first in a particular frame will have a closer relation to the preceding frame in the video sequence, whilst the field to be displayed second will have a closer correlation to the succeeding frame. As previously mentioned, both the interpolated

top field frame (X_T) and the interpolated bottom field frame (X_B) are correlated with the previous frame (X_p) and the next future frame (X_f) such that for each frame in the video sequence four separate correlation values are obtained:

a = correlation (X_T, X_p)
b = correlation (X_B, X_f)
c = correlation (X_T, X_f)
d = correlation (X_B, X_p)

Any suitable metric may be used to measure the correlation, such as peak signal to noise ratio (PSNR), mean square error (MSE) or mean absolute error (MAE). The following table shows the possible results of the correlation check and their interpretation.

Number	Condition	Interpretation
1	a > c and b > d	Field order = top field first
2	a < c and b < d	Field order = bottom field first
3	other conditions	Indeterminate result

It can be seen that result 3 of this frame analysis technique does not produce a definite indication of the field order. To overcome this the method of the present invention applies an averaging technique to assign a field order to those 'indeterminate' frames that applies the principle that it is more probable for a single frame to have the same field order as the surrounding frames. Consequently, an average over a moving window of k frames is taken, with the field order being assigned according to a simple majority across the k frames. An example of this is illustrated in Figure 6, where a sequence of k frames 60 is illustrated, each frame having an indicated field order as determined by an analysis process. It can be seen that for the second frame 62 no determined field order 64 is indicated. However, the remaining k-1 frames are all indicated as bottom field first B. Consequently, according to embodiments of the present invention the second frame 62 is considered

to be bottom field first B also. The inventors of the present application have found that allocating the field dominance according to a 30% percent majority across a window of 25 frames provides robust results. However, it will be appreciated that alternative numbers of frames within the moving window and/or a different majority measure may equally be applied within the scope of the present invention as desired.

Having assigned or determined a field order to all of the frames in the sequence of interest, the determined field order is for each frame is compared to the field order indicated by its metadata. Where the field order given by the metadata does not match the field order indicated by the analysis results then this inconsistency is either immediately flagged to a user or stored in a log file for subsequent retrieval. Alternatively, when such a mismatch occurs the metadata may be automatically amended to match the field order indicated by the results of the analysis process.

The basic method of embodiments of the present invention is illustrated in Figure 5. Consequently in embodiments of the present invention there is provided a robust method of identifying any field order mismatches in a sequence of video frames.

CLAIMS

1. A method of identifying inconsistent field order flags for a sequence of video frames comprising:
- for each frame in the sequence of video frames analysing the frame to make an initial determination of the field order for that frame;
- averaging the initial determination of the field order over a predefined number of most recently analysed frames; and
- determining those frames for which the averaged field order does not match the field order identified by a respective field order metadata item associated with each frame by comparing the averaged field order for each frame to the respective field order metadata item.

2. The method of claim 1, wherein the initial determination of the field order may be indeterminate.

3. The method of claim 2, wherein the indeterminate field order of an analysed frame is replaced by the averaged field order.

4. The method of any preceding claim, wherein the averaging step comprises allocating as the averaged field order the field order of a predetermined proportion of the predefined number of frames having the same field order.

5. The method of claim 4, wherein the predetermined proportion of frames comprises at least 30%.

6. The method of any preceding claim wherein the predefined number of frames over which the averaging step is performed is 25.

7. A computer program arranged to cause a computer to perform the method of any preceding claim.

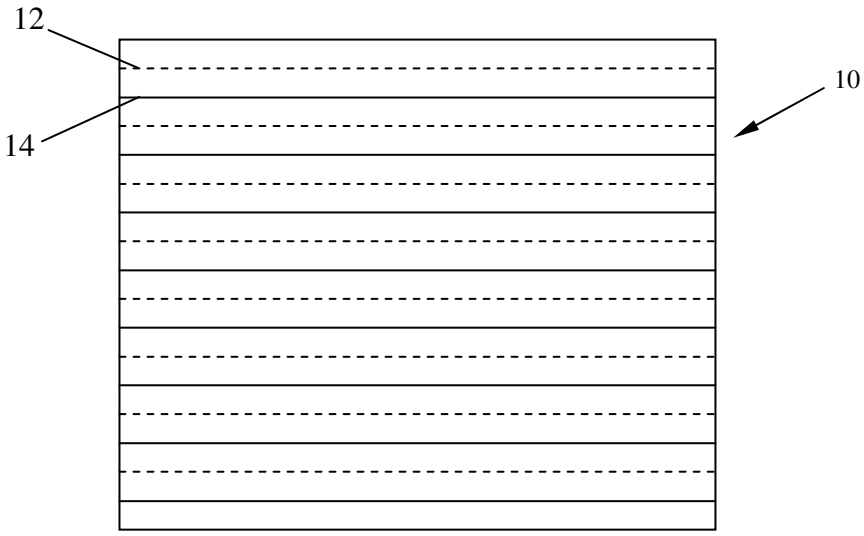


Figure 1

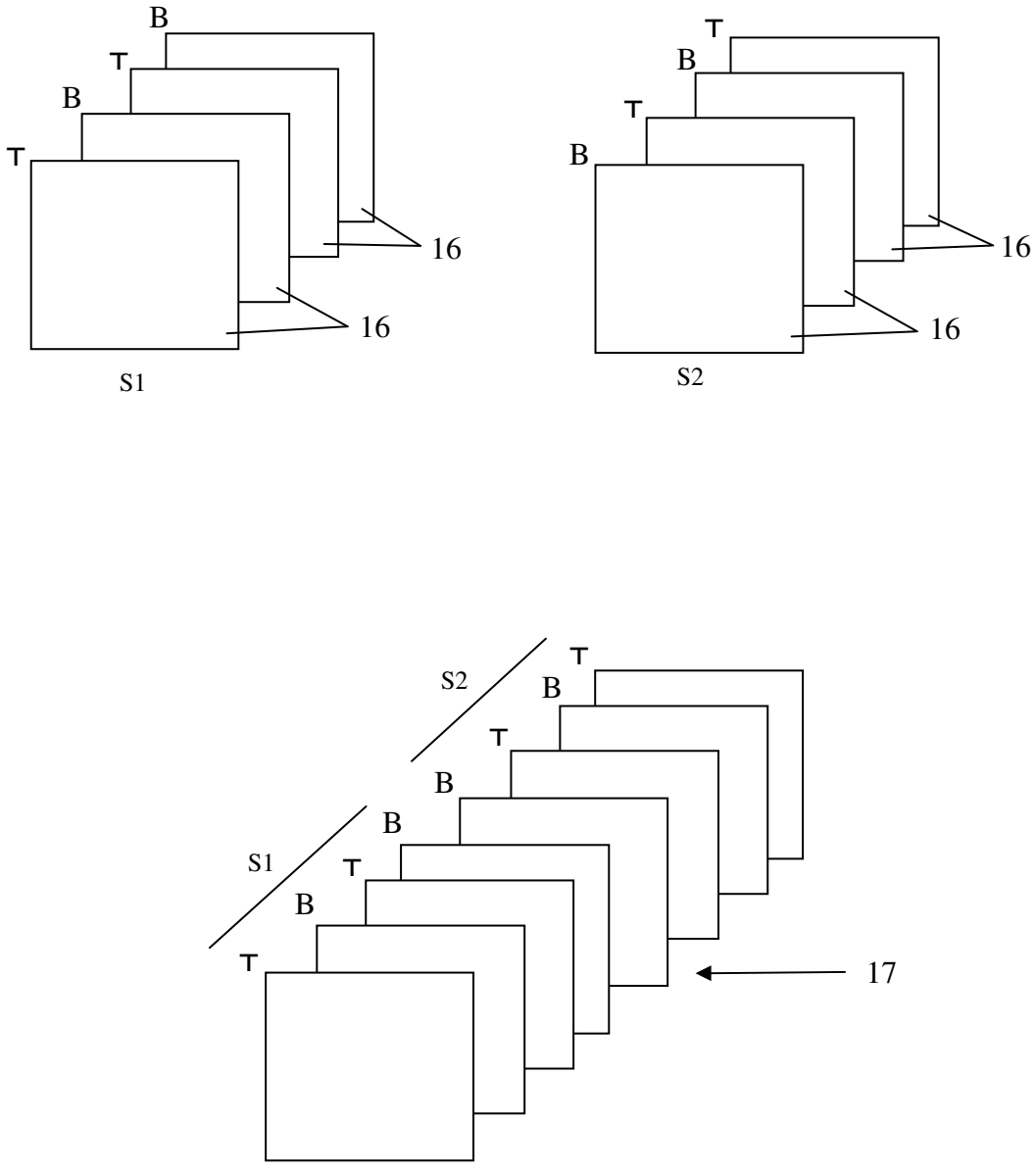


Figure 2

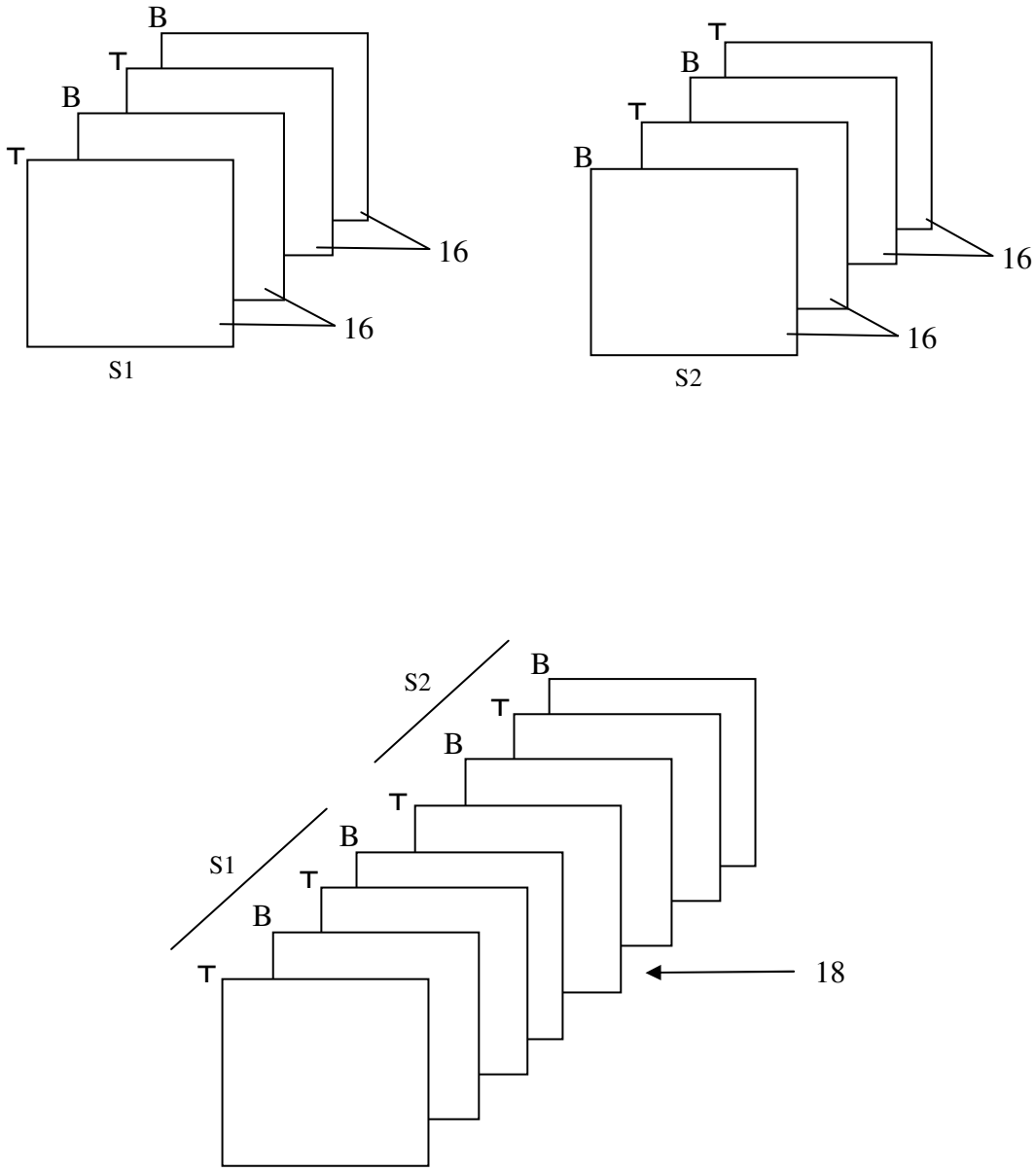


Figure 3

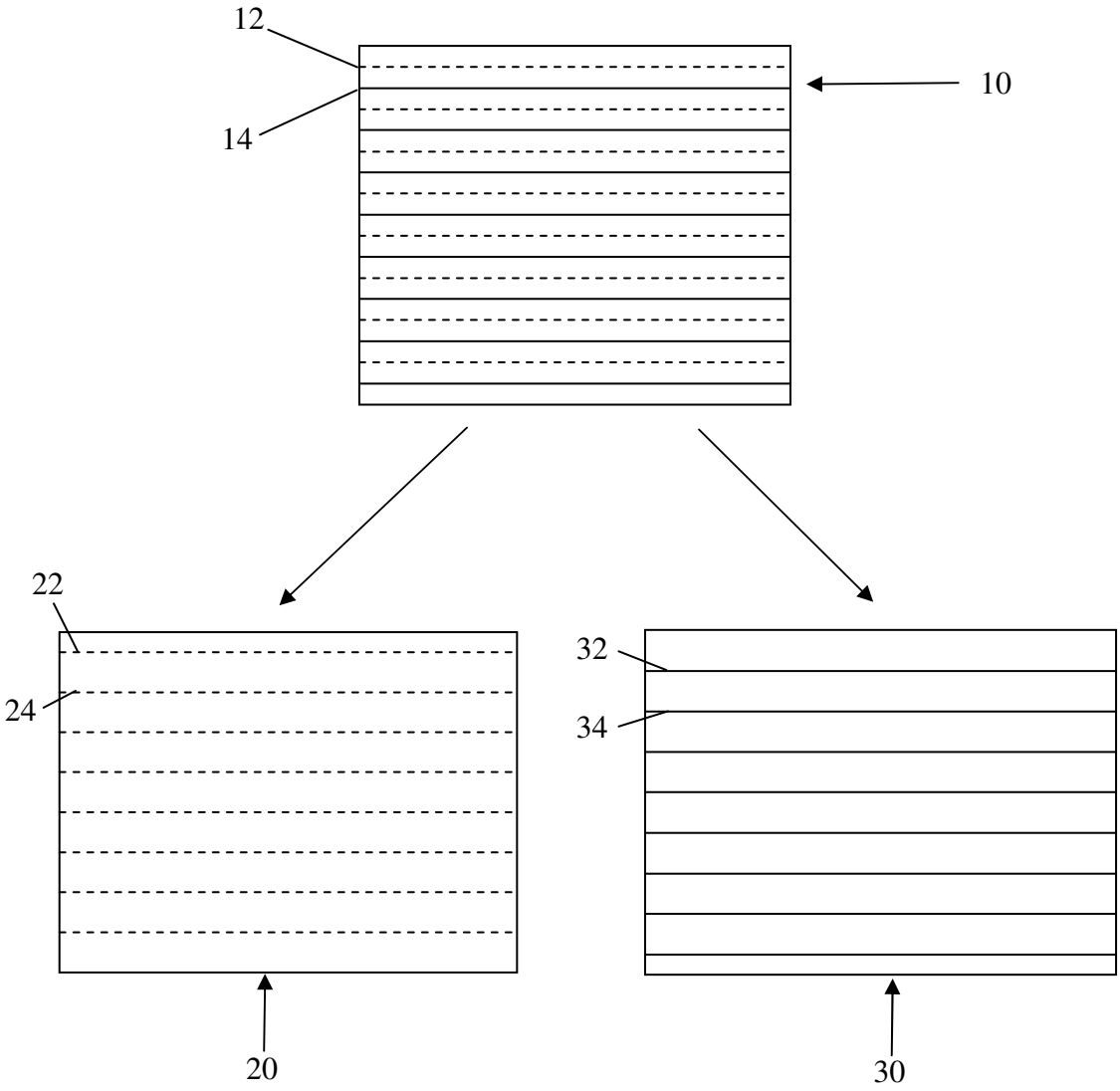


Figure 4

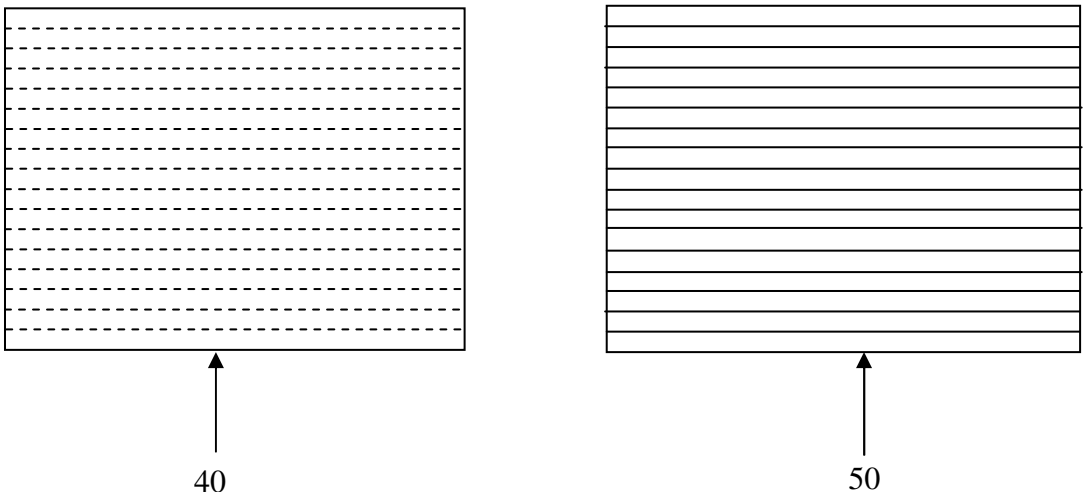


Figure 5

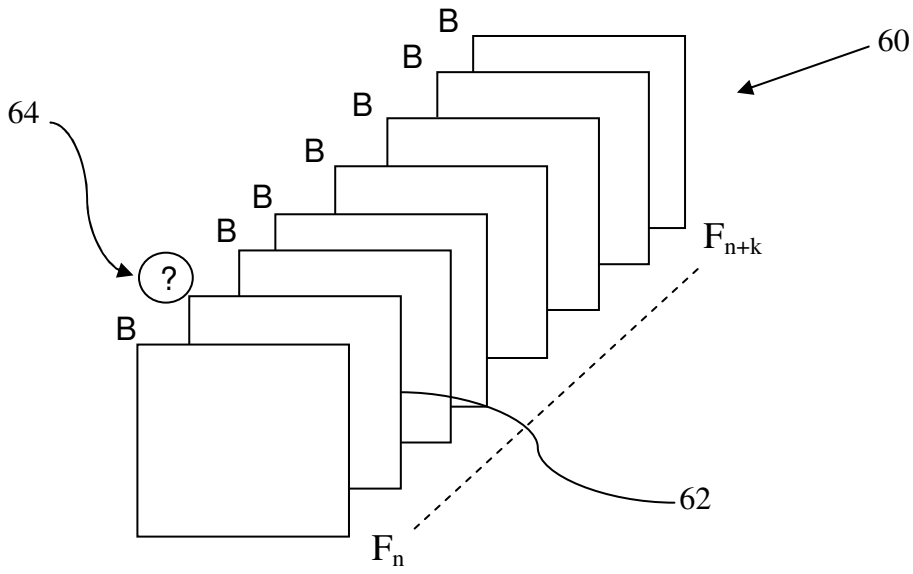


Figure 6

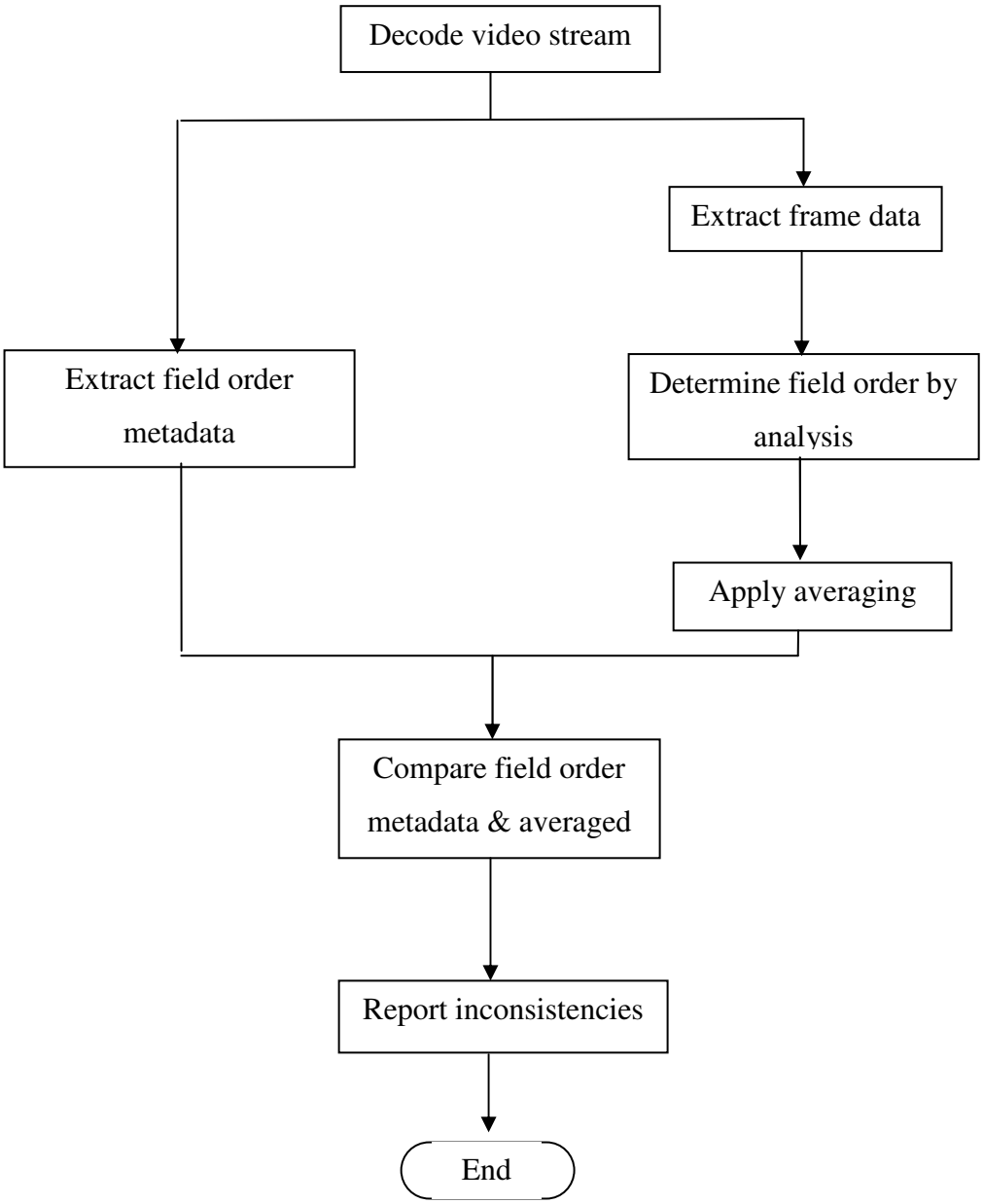

Decode video stream
Extract frame data
Extract field order metadata
Determine field order by analysis
Apply averaging
Compare field order metadata & averaged
Report inconsistencies
End

Figure 7

Appendix G

110497EP/CTW

VIDEO TYPE CLASSIFICATION

ABSTRACT

A video classification method includes detecting pulldown video frames from within a sequence of video frames, for each video frame within said sequence identifying those frames containing inter-field motion, for each frame containing inter-field motion generating a corresponding top field and bottom field, separately correlating the generated top field with a top field of the video frame immediately previous to the frame containing inter-field motion and with a top field of the video frame immediately subsequent to the frame containing the inter-field motion, separately correlating the generated bottom field with a bottom field of the immediately previous video frame and with a bottom field of the immediately subsequent video frame and determining from the outcome of said correlations if the frame containing inter-field motion is a pulldown frame.

Video data may be classified as interlaced, progressive or, particularly where multiple video streams have been edited together, a mixture of both interlaced and progressive video, which is referred to herein as hybrid video. In an interlaced video sequence each frame of the video is made up from two separate fields, one field containing all of the evenly numbered horizontal lines of pixels, referred to as the top field, and the second field containing the even numbered horizontal lines of pixels, referred to as the bottom field. The top and bottom fields represent separate instances in time, i.e. one field is captured at a first instance in time and the second field is captured at a second, subsequent, instance in time. In a progressive video sequence the two fields of each frame belong to the same instance in time.

A particular type of interlaced video is telecine or pulldown video. Telecine is a process by which video material originating from film is converted to an interlaced

format to be displayed on television equipment. As the original film material is generally shot at 24 full frames per second (fps), and which therefore can be considered as progressive image data, the conversion to telecine video requires a frame rate conversion, particularly for NTSC format, since NTSC and PAL video is played at approximately 30 fps (30,000/1,001) and 25 fps respectively. Although it would be possible to simply increase the speed of playback of the original film material, this is generally quite easy to detect visually and audibly, especially for NTSC playback where the increase from 24 fps to approximately 30 fps represents an approximately 25% increase in playback speed. Consequently, a technique is used to increase the number of frames per second displayed that involves inserting one or more extra frames of image data on a repeated basis to increase the total number of frames to be displayed. Generally, this involves generating the extra frames using information from one or more of the original adjacent frames of image data. For NTSC conversion, this is achieved by converting every four frames of image data to their equivalent eight fields (top and bottom field pairs) and then repeating at least two of the individual fields to generate the required number of extra frames. The extra frames generated using duplicated fields are referred to as either pulldown frames or dirty frames. For PAL conversion two additional frames are generated for every twelve original frames to achieve the 24 fps to 25 fps frame rate conversion required.

Reverse telecine is the opposite process in which the pulldown frames are removed and the original 24 fps material is reconstructed. This may be required where the video data is to be displayed on a progressive display that is able to support the 24 fps frame rate or alternatively where the telecine video data is to be compressed, for example prior to data storage or transmission, and is therefore more efficient in terms of the compression to remove the dirty frames since they are redundant by virtue of being generated from image data already present. A problem arises when it is not known what type of video data is being presented as source data to a reverse telecine process. For example, it is normal practice for many television programmes for many different video sources to be edited together, the different

video sources possibly being a mixture of progressive, interlaced or hybrid video. A problem therefore exists in being able to identify the different types of video data present within a source video data stream that is to have a reverse telecine process applied.

According to a first aspect of the present invention there is provided a method of detecting pulldown video frames from within a sequence of video frames, the method comprising: for each video frame within said sequence identifying those frames containing inter-field motion; for each frame containing inter-field motion generating a corresponding top field and bottom field; separately correlating the generated top field with a top field of the video frame immediately previous to the frame containing inter-field motion and with a top field of the video frame immediately subsequent to the frame containing the inter-field motion; separately correlating the generated bottom field with a bottom field of the immediately previous video frame and with a bottom field of the immediately subsequent video frame; determining from the outcome of said correlations if the frame containing inter-field motion is a pulldown frame.

The step of determining the outcome of the correlations may comprise determining the difference between the correlation of the bottom field of the video frame containing inter-field motion with the bottom field of the immediately previous video frame and the correlation of the top field of the video frame containing inter-field motion with the top field of the immediately previous video frame, determining the difference between the correlation of the top field of the video frame containing interfield motion with the top field of the immediately subsequent video frame and the correlation of the bottom field of the video frame containing inter-field motion with the bottom field of the immediately subsequent video frame and when both difference values exceed a predetermined threshold value determining that said video frame containing inter-field motion is a pulldown frame. Additionally, when both difference values do not exceed the threshold value said video frame may be determined to be an interlaced video frame.

The correlation may comprise correlating any one of Peak Signal to Noise Ratio, Mean Absolute Deviation and Sum of Absolute Errors. According to a further aspect of the present invention there is provided a method of classifying a group of video frames, the method comprising: detecting the pulldown frames contained within the group according to the method of the first aspect of the present invention and classifying those frames as pulldown frames, classifying the remaining frames containing inter-field motion as interlaced frames and classifying the non-pulldown and non-interlaced frames as progressive frames; classifying the group of video frames according to a combination of the majority classification of the separate video frames in the group and the presence of known sequences of individual frames. The pattern matching may be applied to the classified frames in a group if the group includes both pulldown frames and progressive frames. Additionally, the pattern matching may comprise identifying the presence of known sequences of progressive and pulldown frames, said known sequences being consistent with telecine video.

Additionally, a group of frames containing more than one known sequence of progressive and pulldown frames may be classified as broken telecine. Embodiments of the present invention are described below, by way of illustrative non-limiting example only, with reference to the accompanying drawings, of which: Figure 1 schematically illustrates a forward telecine process; Figure 2 schematically illustrates the different possible video types and their relative hierarchy that can be classified according to embodiments of the present invention; and Figure 3 schematically illustrates the methodology of embodiments of the present inventions. An example of a conventional telecine process is schematically illustrated in Figure 1. In Figure 1 the telecine process is represented at separate steps, (i-iv). In the first step i) for progressive frames 1-4 are provided from the original source material, which in the example to be discussed as frame rate of 24 fps. Frames 1-4 are intended to be displayed sequentially in order. Step ii) involves generating individual fields 5 from each of the original frames 1-4, such that in the example illustrated each of the original frames is decomposed to a top field and a bottom field. In the particular

example illustrated in Figure 1 it is assumed that the fields are displayed in top field first order, although it will be understood by those skilled in the art that the field order may be the opposite and is not important to the telecine process. Consequently, the four original frames 1-4 are decomposed to eight individual fields 5. At step iii) individual fields 5 are reordered according to a predefined sequence with two of the fields 6, 7 being duplicated, as indicated by the dashed arrows in Figure 1. Consequently, at step iii) there are now ten fields that in the final step iv) are recombined to produce five full frames 8-12, thus resulting in five final frames for each of the original four frames and therefore increasing the frame rates to 30 fps. However, as indicated in Figure 1 the frame 9 generated from the repeated fields 13-14 is composed of fields representing two separate instances in time, A_T B_B, and is therefore likely to give rise to combing artefacts when displayed on a progressive display. This generated frame 9 is therefore referred to as a "dirty" frame.

As an aside, the telecine scheme illustrated in Figure 1 may be referred to 3:3:2:2 pulldown telecine, since the individual fields 5 decompose from the original full frames 1-4 are reproduced following the 3:3:2:2 order to generate the technical individual fields from which the final five frames are generated. Referring to Figure 1, it can be seen that the first three fields at step iii) are drawn from the first frame 1 of the original sequence, the next three fields are drawn from the second frame 2, the next two fields are drawn from the third frame 3, whilst the final two frames are drawn from the fourth frame 4. An alternative scheme is to arrange the individual fields according to a 3:2:3:2 pattern, from which the generic term of 3:2 pulldown for 24 fps to 30 fps telecine is derived. However, using this latter scheme would generate ten fields at step iii) in the following order: A_T A_B, A_T B_B, B_T C_B, C_T, C_B, D_T D_B, from which it can be deduced that in the final frame sequence for every five frames two frames will be "dirty" frames, as opposed to the single dirty frame generated using the 3:3:2:2 scheme.

As previously noted it is also common to perform a reverse telecine process on provided video data to either allow the original progressive film data to be displayed

on compatible progressive displays or to allow efficient compression to occur. This is easily accomplished if it is known that the source data video is in fact a telecine video and what scheme of telecine has been applied to it. However, it is common for the source video data to be made up from a number of separate sources and therefore contain video data of different types. These video types can be arranged in a general hierarchy, as illustrated in Figure 2. As previously noted, generic video data may include either progressive video, interlaced video or hybrid video. The interlaced video can be further subdivided into traditional interlaced, i.e. in which the source material was captured in an interlaced, time differentiated field manner, and telecine or pulldown video, such as 3:2 telecine the type discussed in relation to Figure 1. In addition, the pulldown video can be of a consistent pattern or a broken pattern. If the entirety of the pulldown video segment comprises a single source of the pulldown video such that the pattern of clean and dirty frames and this will be a consistent pattern. In contrast, the segment of pulldown video may include a number of separate sections of pulldown video that have been edited together. Consequently, where the edits have occurred the pattern of clean and dirty frames may change and/or the actual telecine scheme employed may differ between different sections.

It is therefore useful and desirable to determine the different types of video data present either before or during a reverse telecine process. In particular, it is desirable to be able to determine between the traditional interlaced video data and the actual pulldown video data. To accomplish this determination it is therefore necessary to be able to identify the presence of any dirty frames within the video segment, those dirty frames being indicative of the presence of pulldown video.

According to embodiments of the present invention a method for the detection of the video type includes as an initial step detecting the presence of any combing artefacts in individual frames, since the presence of combing artefacts indicates that the frame is either traditional interlaced or pulldown video. A method of determining and quantifying any inter-field motion (which gives rise to the combing artefacts) in a video frame is described in European patent application no.

08251399.5, also filed by the present applicant. This method processes each video frame by taking the top and bottom fields for each frame and interpolating the top and bottom fields to produce interpolated top and bottom field images and subsequently comparing the interpolated top and bottom field images to each other to determine a value representative of the amount of inter-field motion present between the top field and bottom field. The interpolated top field image may be produced by averaging adjacent lines of the top field with a line of the bottom field which is intermediate the adjacent lines of the top field, and the interpolated bottom field image may be produced by averaging adjacent lines of the bottom field image with a line of the top field image that is intermediate the adjacent lines of the bottom field image. Comparison of the interpolated top and bottom field images is performed by subtracting luminance values of the pixels of one of the interpolated images from luminance values of corresponding pixels of the other of the interpolated images to generate a difference domain frame. If the original video frame from which the interpolated top and bottom and field images were generated is a true progressive frame and then there will only be a very small difference between the interpolated top and bottom field images, differing from noise, compression, interpolation, approximation and vertical differences. If a large difference is found over the entirety of the frame then it can be classified as containing inter-field motion arising from either being a traditional interlaced or being telecine or pulldown frame.

The above method of determining the presence or absence of inter-field motion within each frame is merely one applicable method and other known methods for identifying inter-field motion may be used within the scope of embodiments of the present invention. In a subsequent step of the method of the present invention a determination is made as to whether the frame containing the inter-field motion is either interlaced or a "dirty" pulldown frame. The determination is made by performing a correlation between the fields of the current frame under analysis and the fields of both the previous and future frames. Four correlations are calculated as follows:

C1 = correlation (current frame bottom field, previous frame bottom field)
C2 = correlation (current frame top field, previous frame top field)
C3 = correlation (current frame top field, future frame top field)
C4 = correlation (current frame bottom field, future frame bottom field)

If modulus, (C1-C2) or modulus (C3-C4) is greater than a predetermined threshold value then the current frame is considered to have a repeated field, i.e. be a dirty pulldown frame. For example, considering the example illustrated in Figure 1 and taking the final generated frame A/B as the current frame then it can be seen from the corresponding individual fields that C1 = correlation (B_B, A_B), C2 = correlation (A_T, A_T), C3 = correlation (A_T, B_T) and C4 = correlation (B_B, B_B) and therefore the correlation values C1 and C3 will be low, whilst the correlation values C2 and C4 will be high.

Any objective correlation metric may be used, for example PSNR (peak signal to noise ratio), MAD (mean absolute deviation) or SAE (sum of absolute errors). In one embodiment to the present invention the correlation is carried out using PSNR as the correlation metric and if the correlation difference between the fields of successive frames (i.e. the modulus values) is greater than 8 db then the frame is considered to have a repeated field.

To reduce the influence of false positives (i.e. frames incorrectly identified as interlaced or telecine) the frame data is subsequently processed in groups of frames, for example groups of 100 frames. The number of frames per group may be the figure and may be chosen in dependence upon some prior knowledge of the source video data. However, 100 frames for a frame display rate of 25 fps allows the video type information to be provided for every 4 seconds of video and it is unlikely for normal broadcast for edited segments to be of less than 4 second duration. In fact a segment will tend to be longer than this. The classification of each group of frames is based on a combination of a simple majority of individual frame classifications and the outcome of certain pattern matching algorithms. For example, a majority of

frames being classified as being progressive does not necessarily preclude that group from having a pulldown pattern, since progressive frames are a constituent part of a pulldown pattern. However, true interlaced frames and pulldown frames should not be in the same group and in this instance the majority of the two frame types will govern the classification of the group. If a group contains frames being classified as pulldown frames then one or more pattern matching algorithms may be applied to the group of frames to determine if the group can be classified as a pulldown group as a whole. For example, a regular occurrence of four progressive frames followed by a single pulldown frame will be taken as indicative of the 3:2 pulldown pattern illustrated with reference to Figure 1. In this instance, sub-groups of five frames can be analysed and classified and a majority decision based on the classification of the sub-groups may be made for the group as a whole. Alternatively, other known patterns may be looked for, such as 12:2 pulldown. Possible outputs for each group of frames includes progressive, telecine pattern 1 (e.g. ptppp), telecine pattern 2 (e.g. pttpp) and telecine broken (e.g. ptppp, pttpp), the latter indicating the occurrence of an edit within the group.

Advantages of the embodiments of the present invention include the use of only immersed immediate neighbours to the frame of interest in analysing if that frame is a pulldown frame or not. By using only the immediate neighbours to a frame under analysis, as opposed to a series of frames, any spatial or temporal variations across a series of frames do not unduly influence the outcome of the determination, which such variations would influence the outcome if a larger series of frames were used. Similarly, the classification of each group of frames is processed independently and no assumptions are made based on the results for previous groups. This particularly increases the robustness of the method when applied to hybrid video sequences and allows any change in video type due to editing to be easily detected.

CLAIMS

1. A method of detecting pulldown video frames from within a sequence of video frames, the method comprising:

- for each video frame within said sequence identifying those frames containing inter-field motion;
- for each frame containing inter-field motion generating a corresponding top field and bottom field;
- separately correlating the generated top field with a top field of the video frame immediately previous to the frame containing inter-field motion and with a top field of the video frame immediately subsequent to the frame containing the inter-field motion;
- separately correlating the generated bottom field with a bottom field of the immediately previous video frame and with a bottom field of the immediately subsequent video frame;
- determining from the outcome of said correlations if the frame containing inter-field motion is a pulldown frame.

2. The method of claim 1, wherein the step of determining the outcome of the correlations comprises:

- determining the difference between the correlation of the bottom field of the video frame containing inter-field motion with the bottom field of the immediately previous video frame and the correlation of the top field of the video frame containing inter-field motion with the top field of the immediately previous video frame;
- determining the difference between the correlation of the top field of the video frame containing interfield motion with the top field of the immediately subsequent video frame and the correlation of the bottom field of the video frame containing inter-field motion with the bottom field of the immediately subsequent video frame; and
- when either difference values exceed a predetermined threshold value determining that said video frame containing inter-field motion is a pulldown frame.

3. The method of claim 2, wherein when both difference values do not exceed the threshold value determining said video frame to be an interlaced video frame.

4. The method of any preceding claim, wherein the correlation comprises correlating any one of Peak Signal to Noise Ratio, Mean Absolute Deviation and Sum of Absolute Errors.

5. A method of classifying a group of video frames, the method comprising:

- detecting the pulldown frames contained within the group according to the method of claim 1 and classifying those frames as pulldown frames, classifying the remaining frames containing inter-field motion as interlaced frames and classifying the non-pulldown and non-interlaced frames as progressive frames;
- classifying the group of video frames according to a combination of the majority classification of the separate video frames in the group and the presence of known sequences of individual frames.

6. The method of claim 5, wherein pattern matching is applied to the classified frames in a group if the group includes both pulldown frames and progressive frames.

7. The method of claim 6, wherein the pattern matching comprises identifying the presence of known sequences of progressive and pulldown frames, said known sequences being consistent with telecine video.

8. The method of claim 7, wherein a group of frames containing more than one known sequence of progressive and pulldown frames is classified as broken telecine.

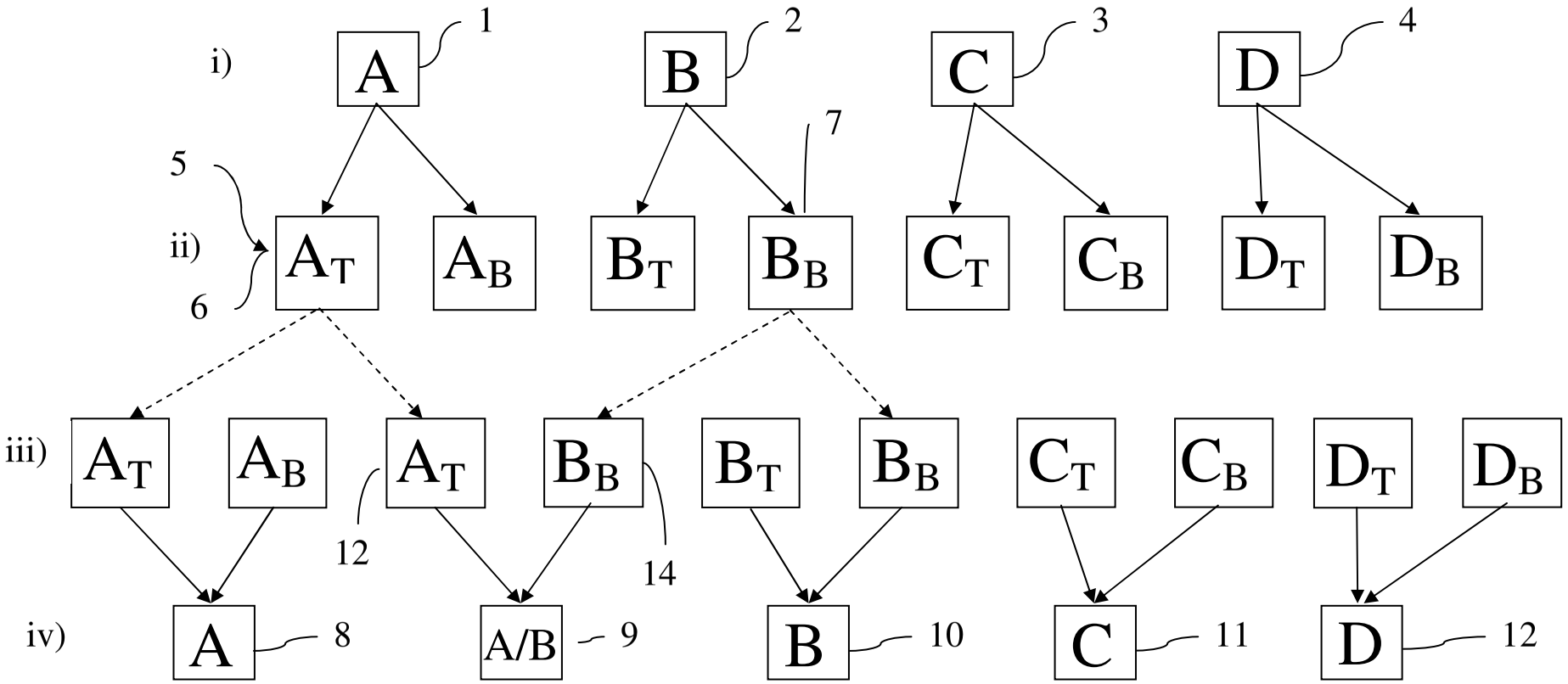


Figure 1

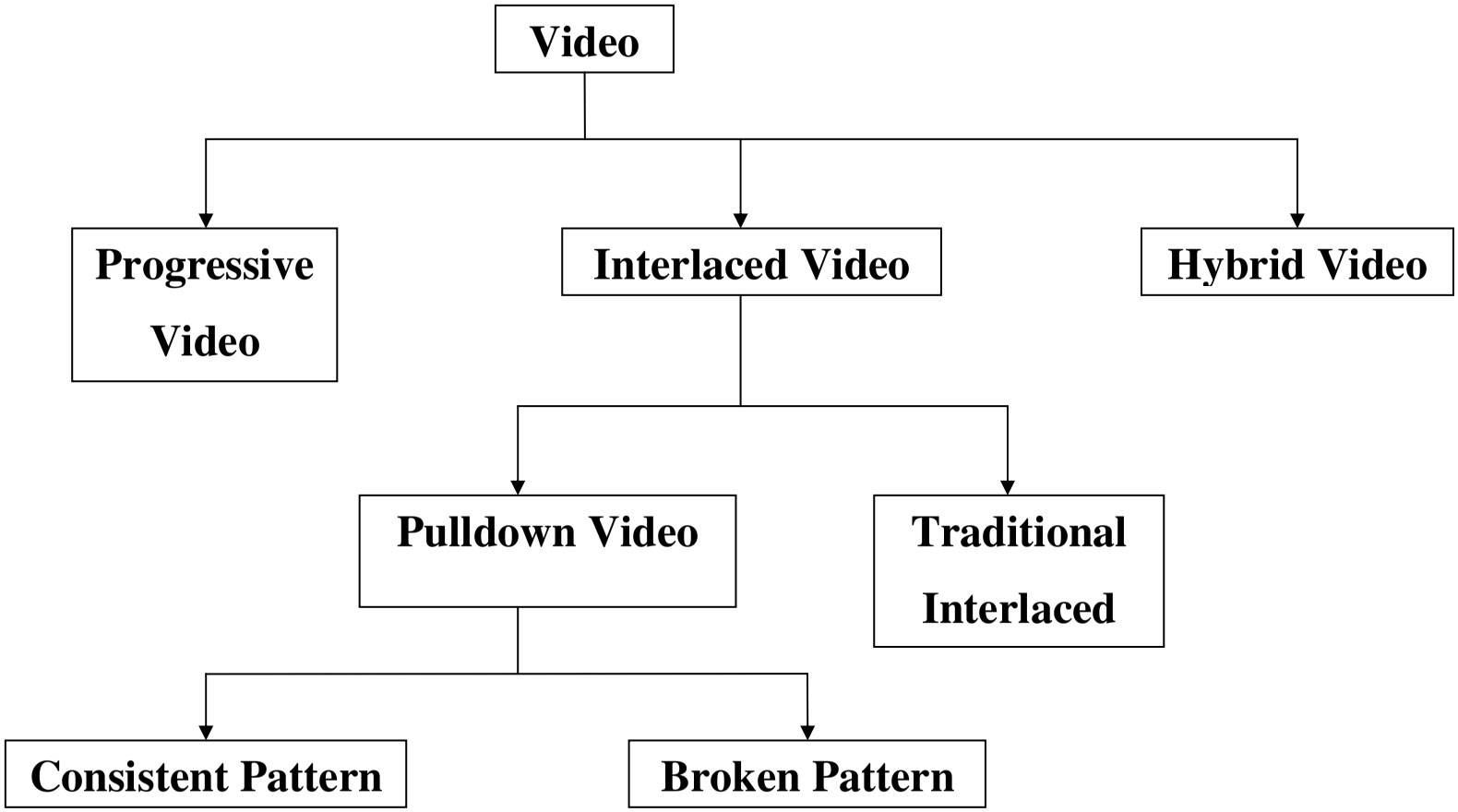

Video
Progressive Video
Interlaced Video
Hybrid Video
Pulldown Video
Traditional Interlaced
Consistent Pattern
Broken Pattern

Figure 2

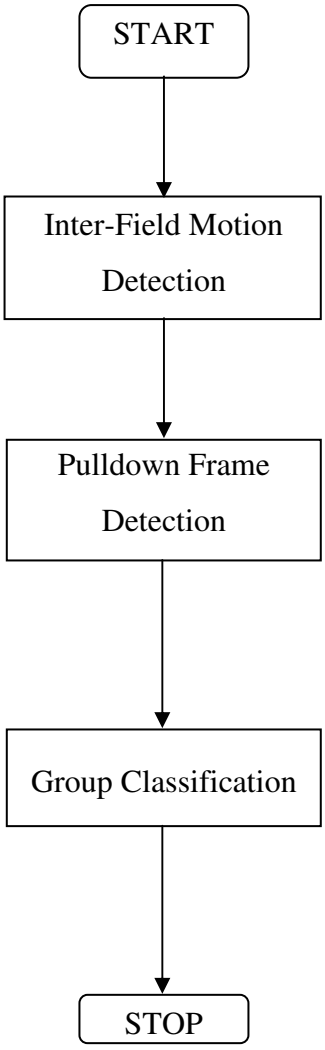

START
Inter-Field Motion Detection
Pulldown Frame Detection
Group Classification
STOP

Figure 3